DATA PRIVACY, CS295

RELEASING DIFFERENTIALLY PRIVATE EDGE COUNTS OF COMPLEX NETWORKS

David W. Landay, Samson C. Durst
University of Vermont
Comp. Systems and Data Sci.

December 9, 2019

Abstract

We explore a Lipschitz extension, as described in [1], for releasing node-differentially private edge counts over a well-known, real-world, and small graph called the *karate* network. Implementing algorithm **4.1** from [1] we demonstrate that an analyst can return an edge count under bounded sensitivity for a small network, which satisfies ϵ-node differential privacy. Furthermore, we show that releasing a noisy edge count under **4.1** guarantees the count is more accurate than one returned under global sensitivity for a small increase in privacy cost.

Introduction

In the era of *Big-Data*, platforms like Facebook, Twitter, and Instagram have become a pervasive and important part of our lives. Often, these social networks are used by scientists in analytical studies, by advertising agencies in marketing campaigns, and by millions of everyday users as a means to communicate with one another. In general, when it is not being abused, these studies have the potential to drastically improve our lives. However, these services also allow for the spread of misinformation, and for bad actors to learn personally identifiable information about users **and** their neighbors in the network. As a result, protecting an individual's data from leaving a network must become a top priority and obligation for analysts; one that is difficult to satisfy.

In the majority of networks, achieving differential-privacy – in order to release edge counts, mean queries, and other graph related inquiries privately – is hard. A traditional data-set is often described as a table of rows and columns, in which rows represent indivuals, and columns represent personal characteristics about them. Rows are independent of one another, and cannot necessarily reveal information about any other row in the set. Graphs, or networks, do not abide by this convention because they show the relationships between individuals by design. Hence, releasing information from a network may not ensure privacy for all individuals existing within it if privacy is framed in the same way as it is for data-tables. A new paradigm must be introduced to address this issue.

Node-DP & Edge-DP

In [1], the authors explain that **node-differential privacy** (node-DP) and **edge-differential privacy** (edge-DP) are two new conventions that are necessary for releasing differntially-private queries on networks. Specifically, they help define what a neighboring dataset is in the context of graphs and network data. Edge-DP is satisfied if the removal or addition of one edge in the network is not revealed. Two graphs G and G' are said to be *edge neighbors* (neighboring datasets) if they differ by exactly one edge. An algorithm satisfies ϵ-edge-DP if the concept of edge neighbors is preserved. Similarly, Node-DP is satisfied when the inclusion or exclusion of a node and all of it's adjacent edges are protected from an algorithms output. Two Graphs are said to be *node neighbors* if one can be constructed from the other by the removal of a node and its adjacent edges. An algorithm is said to be ϵ-node-DP if the concept of node neighbors is satisfied. As suggested previously, it is often harder to satisfy ϵ-node-DP for queries that have unbounded sensitivity.

In this paper, we focus on how to release the number of edges in a network in a differentially private way. But, this is not a simple task, as the sensitivity of such a query can be unbounded. A simple 1-dimensional query over graphs, like counting the number of nodes, still allows analyst to use

traditional means, like the Laplacian Mechanism, to release a differentially-private answer scaled to global sensitivity. This is because the removal of one edge from a network will not change the total number of nodes, and removing one node from the network will only change the total number of nodes by exactly one. Hence, releasing the node count with the laplace mechanism scaled to a global sensitivity of 1 satisfies both ϵ-node-DP and ϵ-edge-DP.

When it comes to releasing the number of edges, the same logic does not apply. It does not satisfy ϵ-node-DP under global sensitivity because the removal of one node could remove an unbounded number of edges. Furthermore, if we look at the true degree distribution, we find that choosing an upper bound on the number of edges will have an effect on privacy. If we select an upper bound $k = |V|$, where V is the set of vertices (nodes) in the graph, then we achieve good privacy, but destroy the utility of the released query answer. The question becomes: Is there a good bound on sensitivity such that the edge count can be released and satisfy ϵ-node-DP?

Graph Clipping

The first thought to consider when trying to solve this problem, is if there is a good global bound on sensitivity for releasing the edge count of a small network. This of course would be the simplest method and would yield results while giving the least amount of thought to the problem. We can achieve this by specifying an arbitrary, or best estimate, of the upper limit on degree for any node in the network. We can then clip a random subset of adjacent edges to any node with a degree higher than the proposed upper limit. Finally, we can release a noisy count of the number of edges of the clipped graph by applying the Laplace Mechanism with sensitivity scaled to the proposed clipping parameter, and given some epsilon. On certain real-world networks, namely ones that are large, this can work well more often than if the network were small because the removal of one node is less likely to adversely affect the count of the total number of edges. Within a smaller network, if the degree distribution is heavy-tail, removing one node that contains many edges will require noise with a high global sensitivity, and will likely release terribly inaccurate results.

Methods

The Karate Network

The Karate network is a widely used benchmark for studies of small-scale social dynamics. It was first introduced by Wayne W. Zachary in [2]. It describes the 78 links between pairs of the 34 members (nodes) of a karate club, who met outside of club meetings. Wayne studied the relationships

of the club members for three years, during which a conflict arose between "John A", the club's administrator, and an instructor "Mr. Hi". In **Fig. 1**, the two sets of colored nodes represent the two factions that formed as a result of the conflict.

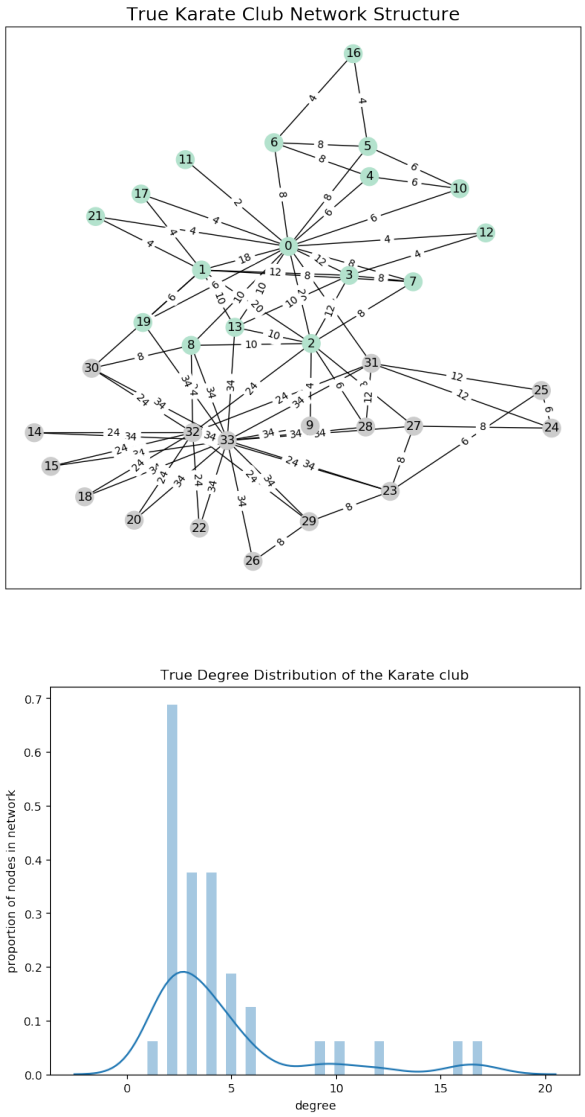

Figure 1: (Top) The two factions of Zachary's Karate Network. There are 78 edges among 34 nodes. **(Bottom)** The degree distribution of the karate network.

As one can see, there are two or more highly connected nodes in the karate network. Removing these could reduce the utility of an edge count query utilizing the graph clipping method described above.

Flow-Graph Lipschitz extension

If releasing the queries based on global sensitivity does not provide us with useful results for smaller graphs, what will? In short, one answer is to use an iteration of the low-graph Lipschitz extensions defined in section 4.2 of [1]. The overall goal of this method is to improve the accuracy of the noisy edge counts released for smaller networks, while avoiding the use of a global sensitivity. The most important piece of this algorithm is the sensitivity required to calculate the noise to release the edge counts.

The node-DP algorithm for releasing $f_e(G)$ (edge count) stipulates the following rules: The analyst supplies ...

- a proposed privacy cost ϵ,
- capacity, D, which represents the upper bounds on flow over a network.
- n nodes
- and a graph G

As described in lemma 4.3 in [1], a threshold τ is defined as $\frac{n \cdot \ln n}{\epsilon}$. The value for τ will determine whether we can release $f_e(G)$ with Laplacian noise, make further calculations on $f_e(G)$ before we can release a noisy edge count. Before continuing, we want to disclose the fact that the authors of [1] do not prove the sensitivity of the algorithm in their paper, but they do outline how τ is assigned in the lemma.

We calculate $\hat{e_1} = f_e(G) + \mathcal{L}(\frac{2n}{\epsilon})$, where $\mathcal{L}$ denotes the Laplace mechanism, and check whether it is above 3τ. If it is not, we are safe to release $\hat{e_1}$ by lemma 4.3. Otherwise, we compute the maximum flow over the network, v_{fl}, and release $\hat{e_2} = \frac{v_{fl}(G)}{2} + \mathcal{L}(\frac{2D}{\epsilon})$, where D is the maximum flow capacity proposed by the analyst. The purpose for scaling the sensitivity by $2 \cdot D$ is a response to returning answers to queries on small networks, which are greatly impacted by removing one important node. Instead of setting a global sensitivity to the maximum possible degree of a node, we scale the sensitivity to D; the theoretical number of edges a node can traverse when it visits another node in the network. We can consider this a proxy for the total impact on the network that a particular node will have if it removed. This scaling can minimize the problem described above, and avoids returning inaccurate answers. See algorithm 1 in [1].

Flow Graphs

A flow graph, also referred to as a transposition network, is a directed graph where each node has some flow, and there exists an edge attribute capacity, defined by the analyst, that describes the amount of flow that passes between two nodes. For our application, it is appropriate to define capacity as the maximum total number of edges that can be traversed when we measure the flow between two nodes. As an example, in **Fig. 1** we have constructed a directed flow graph from the un-directed karate network, and the flow values for every pair of adjacent nodes are plotted along each edge. The global maximum flow over the network, Δv_{fl}, is bounded by $\Delta v_{fl} \leq 2D$ as described by lemma 4.1 in [1]. This is because at most every node could be connected to every other node, in which case we expect the max flow between any two nodes to be $2 \cdot D = 2 \cdot f_e(G) = n$.

Formally, a flow graph G is a directed graph; $G = (V,E)$; $(u,v) \in E \implies (v,u) \notin E$. In [1], we construct a directed graph from the undirected network of interest. Flow is measured between a selected source node s =, and a sink node t that is reachable from s. The flow graph G with parameter D, source s, and sink t is a directed graph on nodes $V_s \cup V_t \cup \{s,t\}$, where V_s and V_t are identical copies of V; the set of nodes in the un-directed Karate network. This satisfies the following properties of a flow graph:

- For any non-source and non-sink node, the input flow is equal to output flow.
- For any edge E_i in the network, $0 \leq v_{fl}(E_i) \leq D(E_i)$.
- Total flow out of the source node is equal total to flow in to the sink node.
- Net flow over the edges follows skew symmetry i.e. $v_{fl}(u,v) = -v_{fl}(v,u)$.

Results

Below, we compare the accuracy of releasing differentially private edge counts using the graph clipping technique, and the Flow-Graph Lipschitz extension. We should point out that we achieved more interpretable and accurate answers, when we released edge counts under bounded sensitivity with $\frac{\Delta v_{fl}}{4}$. The two sets of box-plots are an attempt to highlight why we are limited by global sensitivity when releasing edge counts of small networks.

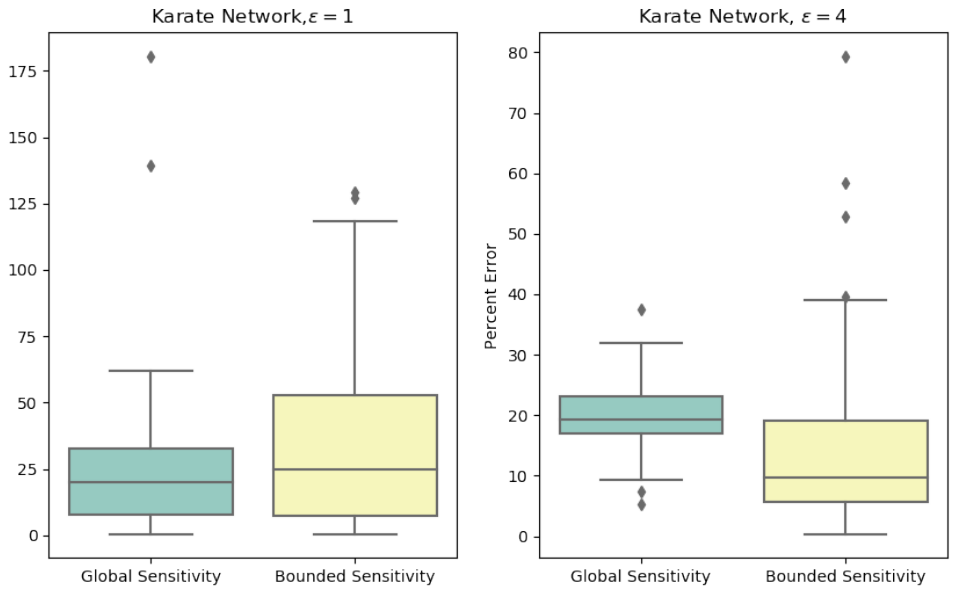

Figure 2: Comparison of releasing noisy edge counts that satisfy -node-DP. The box plots labeled "Global Sensitivity" are queries released using graph clipping. The box plots labelled "Bounded Sensitivity" are queries released using the Flow-Graph Lipschits extension. Each plot shows results for a different privacy budget , and represents a sample of 50 queries generated by the same random seeds.

Discussion & Conclusion

The results in **Fig. 2** demonstrate that a release of edge count queries using the graph clipping technique has consistent error rates for increased privacy budgets over a small network. Furthermore, the results show that releasing accurate, differentially private, edge count queries using the Flow-Graph Lipschitz extension will depend heavily on the amount of privacy, ϵ, that an analyst is willing to give up. We can conclude that better answers can almost always be guaranteed with this algorithm, given that the proper ϵ and capacity D are chosen.

Future Work

The major limitation of the Flow-Graph Lipschitz extension technique is that the analyst must propose the max. capacity parameter D. Currently, the best one can do is use a differentially private release of the number of nodes to estimate an upper bound on D, but are limited by the need for prior knowledge of the network structure. However, if the problem can be framed as a series of sensitivity-1 queries, then the sparse vector technique [3] can be applied to find an upper bound on D, which will release an accurate answer to an edge-count query. We will benefit from the lower scaling

of the sensitivity over the noisy answer that will be released.

Bibliography

[1] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing Graphs with Node Differential Privacy. In Amit Sahai, editor, *Theory of Cryptography*, volume 7785, pages 457–476. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[2] Wayne W. Zachary. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4):452–473, December 1977.

[3] Min Lyu, Dong Su, and Ninghui Li. Understanding the Sparse Vector Technique for Differential Privacy. *arXiv:1603.01699 [cs]*, September 2016. arXiv: 1603.01699.