

## MIT Open Access Articles

*Smooth Anonymity for Sparse Graphs*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Epasto, Alessandro, Esfandiari, Hossein, Mirrokni, Vahab and Munoz Medina, Andres. 2024. "Smooth Anonymity for Sparse Graphs."

**As Published:** 10.1145/3589335.3651561

**Publisher:** ACM

**Persistent URL:** <https://hdl.handle.net/1721.1/155161>

**Version:** Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

**Terms of Use:** Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



# Smooth Anonymity for Sparse Graphs

Alessandro Epasto

Google

New York City, New York, USA

aepasto@google.com

Vahab Mirrokni

Google

New York City, New York, USA

mirrokni@google.com

Hossein Esfandiari

Google

London, UK

esfandiari@google.com

Andres Munoz Medina

Google

New York City, New York, USA

ammedina@google.com

## ABSTRACT

In this work, we aim to manipulate and share an entire sparse dataset with a third party privately. As our first main result, we prove that *any* differentially private mechanism that maintains a reasonable similarity with the initial dataset is doomed to have a very weak privacy guarantee. Next, we consider a variation of  $k$ -anonymity, which we call smooth- $k$ -anonymity, and design a simple large-scale algorithm that efficiently provides smooth- $k$ -anonymity. We further perform an empirical evaluation and show that our algorithm improves the performance in downstream machine learning tasks on anonymized data.

## CCS CONCEPTS

• Security and privacy → Data anonymization and sanitization; Privacy protections; • Theory of computation → Theory of database privacy and security.

## KEYWORDS

Privacy;  $k$ -Anonymity; Differential Privacy; Sparse Graphs

### ACM Reference Format:

Alessandro Epasto, Hossein Esfandiari, Vahab Mirrokni, and Andres Munoz Medina. 2024. Smooth Anonymity for Sparse Graphs. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion)*, May 13–17, 2024, Singapore, Singapore. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3589335.3651561>

## 1 INTRODUCTION

In this work, we study a situation where we intend to share an entire dataset, without violating user privacy. The dataset might be used by the public for several different purposes. Hence, to measure the accuracy regardless of the downstream task, we use a general-purpose metric to measure the similarity of the initial dataset with the shared dataset. To measure privacy there is a large body of work. However, at a high level, there are two distinct approaches to quantifying privacy, *differential privacy* and *k-anonymity*.

Differential privacy is a property of a data processing algorithm and it ensures that small changes in input lead to minimal changes in the output. All differentially private algorithms are randomized, and the uncertainty introduced by the randomization provides a layer of protection. On the other hand,  $k$ -anonymity is a property of the dataset. To make a dataset  $k$ -anonymous one either generalizes or removes data that is identifiable, so that in the final dataset any information is shared by at least  $k$  distinct users.

In this work, we prove that sharing sparse binary matrices with differential privacy guarantees is infeasible (See Theorem 3.1). Roughly speaking, we prove that any differentially private algorithm either provides a very weak privacy guarantee or significantly changes the dataset.

On the other hand, making a dataset  $k$ -anonymous while preserving utility optimally is an NP-hard problem [1]. Current approximation algorithms offer the guarantee of removing at most  $O(\log(k))$  times more elements than that of an optimal solution, however, such a bound is vacuous when the optimal solution has to remove a constant fraction of the dataset. In those cases, the algorithm that just returns a null dataset achieves the same guarantee.

Here, as our second main result, we study a variant of  $k$ -anonymity (called smooth- $k$ -anonymity). We provide a polynomial-time approximation algorithm for smooth- $k$ -anonymity in binary matrices and in theory improve the approximation guarantees of the state of the art results for  $k$ -anonymization. In the binary matrix representation, each row represents the data of one user and each column corresponds to a feature, and if the user  $u$  has the feature  $f$ , element  $(u, f)$  in the matrix is 1.

**Related works:** One of the first techniques for anonymizing data sets was  $k$ -anonymity [10]. This notion was intended for tabular data where each row corresponds to a user and each column corresponds to a particular feature. Other works have improved upon this definition by enforcing other restrictions such as requiring  $l$ -diversity [8] or  $t$ -closeness [6], on top of  $k$ -anonymity. The choice of quasi-identifiers is crucial since an attacker with just a small amount of information about a user could de-anonymize a dataset [9]. Differential privacy has been effectively applied for statistics release [3] and empirical risk minimization [2] among many other scenarios. The vast majority of differential privacy examples require the mechanism to output a summarized version of the data: a statistic or a model in the case of risk minimization.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

WWW '24 Companion, May 13–17, 2024, Singapore, Singapore

© 2024 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0172-6/24/05.

<https://doi.org/10.1145/3589335.3651561>

## 2 SETUP

Given the equivalence of binary matrices and bipartite graphs, for ease of notation we mostly use graph theoretical terminology to describe our work. We assume we are given a bipartite graph, where one set of nodes corresponds to users and another set of nodes corresponds to features. This is a common modeling step, for instance in location analysis applications the features may represent places visited; in social network modeling, the features may represent interests shared by different users; and so on.

Let  $U = \{u_1, \dots, u_n\}$  denote a set of users and  $F = \{f_1, \dots, f_m\}$  a set of features. Throughout the paper we use  $n$  and  $m$  as the  $|U|$  and  $|F|$ , respectively. The edge set  $E$  of the graph is defined as follows, given  $u \in U$  and  $f \in F$ , we say  $e = (u, f) \in E$  if user  $u$  is associated with item  $f$ . We denote this graph by  $G = (U \cup F, E)$ . Let  $\mathbb{G}$  denote the space of all bipartite graphs over  $U \cup F$ , a mechanism  $\mathcal{M}: \mathbb{G} \rightarrow \mathbb{G}$  is a (possibly randomized) function that maps  $G = (U \cup F, E)$  to another graph  $G' = (U \cup F, E')$  with the same set of nodes but with possibly different edges. Given two sets  $A$  and  $B$  we denote their symmetric difference by  $A \oplus B$ .

We now introduce the different notions of privacy we will be using throughout the paper.

**Definition 2.1.** *Edge differential privacy.* We say a randomized mechanism  $\mathcal{M}$  preserves  $\epsilon$ -edge differential privacy if for any two graphs  $G = (U \cup F, E)$  and  $G' = (U \cup F, E')$  such that  $|E \oplus E'| = 1$  the following holds for all  $A \in \mathbb{G}$ :

$$P(\mathcal{M}(G) \in A) \leq e^\epsilon P(\mathcal{M}(G') \in A),$$

Node differential privacy is similar to the edge differential privacy with the difference that it changes a node instead of an edge.

We will consider graphs in  $\mathbb{G}$  with fixed node sets  $U \cup F$ , and varying edge sets. Let  $G = (U \cup F, E) \in \mathbb{G}$  be one such graph, notice that the graph is identified by  $E$ . For a given edge set  $E$ , let  $F_u(E) = \{f \in F: (u, f) \in E\}$  be the items associated with  $u$  in the set edge set  $E$ . Notice we can then partition users into equivalence classes. Formally, let

$$C_u(E) = \{u' \in U | F_{u'}(E) \equiv F_u(E)\}.$$

Now we are ready to formally define  $k$ -anonymity by suppression.

**Definition 2.2** ( $k$ -anonymization and  $k$ -anonymization by suppression). A mechanism  $\mathcal{M}$  is  $k$ -anonymous if for any graph  $G = (U \cup F, E) \in \mathbb{G}$ ,  $\mathcal{M}(G) = (U \cup F, E')$  satisfies:

- (1) For every  $u \in U$ ,  $|C_u(E')| \geq k$ .

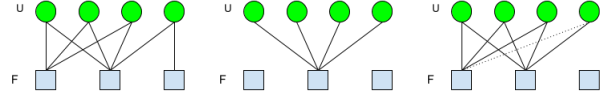
$\mathcal{M}$  is  $k$ -anonymous by suppression if we also have  $E' \subset E$

That is the set of items associated with each user in the output, is the same of that of at least  $k$  users. Also, in  $k$ -anonymity with suppression the output set of edges  $E'$  needs to be a subset of  $E$ . Notice that with a  $k$ -anonymous output an adversary can only distinguish a user up to a set of  $k$  different people.

Finally, we introduce our variant of the above definition.

**Definition 2.3** (smooth- $k$ -anonymity). A mechanism  $\mathcal{M}$  is smooth- $k$ -anonymous if for any graph  $G = (U \cup F, E) \in \mathbb{G}$ ,  $\mathcal{M}(G) = (U \cup F, E')$  satisfies:

- (1) For every  $u \in U$ ,  $|C_u(E')| \geq k$ .
- (2) For every  $u \in U$ , and every  $f \in F$ ,  $(u, f) \in E'$  implies  $|\{u' \in C_u(E'): (u', f) \in E'\}| \geq |C_u(E')|/2$



**Figure 1: From left to right: original graph,  $k$ -anonymous with suppression graph and smooth- $k$ -anonymous graph.**

This definition is very similar to Definition 2.2. The main difference between the definitions is that a smooth- $k$ -anonymous mechanism is only allowed to add edges to the output if, for each equivalence class of users and each item connected to them, the majority of such edges belong to the original graph. In Figure 1 we depict the difference between our smooth- $k$ -anonymous and  $k$ -anonymity with suppression.

We conclude this section by defining the utility measure of a mechanism. In order for a mechanism to be useful it should preserve as much as possible of the graph structure. In this paper we measure this by the Jaccard similarity of two graphs.

**Definition 2.4.** Given two graphs  $G = (U \cup F, E)$ ,  $G' = (U \cup F, E')$  we denote the Jaccard similarity of them by  $J(G, G') := \frac{|E \cap E'|}{|E \cup E'|}$ .

## 3 HARDNESS FOR DIFFERENTIAL PRIVACY

Here as one of the main results we answer the following question: "Is it possible to design an  $\epsilon$ -differential privacy mechanism with a small  $\epsilon$  that guarantees the output to be similar to the input?". The following theorem rules out the existence of such a mechanism.

**THEOREM 3.1.** Let  $\mathcal{M}$  be an arbitrary mechanism that satisfies  $\epsilon$ -edge differential privacy. Let  $\alpha$  be a parameter such that for any input graph  $G = (U \cup F, E)$ , we have  $\alpha \leq E[J(\mathcal{M}(G), G)]$ . We have  $\epsilon \in \Omega(\log(\alpha^2 nm))$ .

**PROOF.** To prove this first we define a policy  $\overline{\mathcal{M}}(G)$  based on  $\mathcal{M}(G)$ , such that  $\overline{\mathcal{M}}(G)$  is

- an  $\epsilon$ -differentially private mechanism,
- $E[J(G, \overline{\mathcal{M}}(G))] \geq \frac{\alpha}{2}$ , when  $|G| \geq l = \lfloor (nm)^{0.9} \rfloor$ , and
- $|\overline{\mathcal{M}}(G)| \leq \frac{2(l+1)}{\alpha}$ .

The third property bounds the range of  $|\overline{\mathcal{M}}(G)|$  and allows us to analyze  $\overline{\mathcal{M}}(G)$  and bound  $\epsilon$ . We define policy  $\mathcal{M}(G)$  based on  $\mathcal{M}(G)$  as follows:

- If  $|\mathcal{M}(G)| > \frac{2(l+1)}{\alpha}$  then  $\overline{\mathcal{M}}(G)$  is set to empty graph,
- otherwise  $\overline{\mathcal{M}}(G) = \mathcal{M}(G)$ .

Note that  $\overline{\mathcal{M}}(G)$  can be exactly calculated given  $\mathcal{M}(G)$ , hence  $\overline{\mathcal{M}}(G)$  is an  $\epsilon$ -differentially private policy as well. Moreover, note that when  $|\mathcal{M}(G)| > \frac{2(l+1)}{\alpha}$  we have

$$J(G, \mathcal{M}(G)) = \frac{|G \cap \mathcal{M}(G)|}{|G \cup \mathcal{M}(G)|} \leq \frac{|G \cap \mathcal{M}(G)|}{|\mathcal{M}(G)|} < \frac{\alpha}{2} \frac{|G|}{l+1}. \quad (1)$$

Hence, for a graph  $G$  with  $|G| \leq l+1$  we have

$$\begin{aligned} E[J(G, \overline{\mathcal{M}}(G))] &= E[J(G, \mathcal{M}(G))] - E[J(G, \mathcal{M}(G)) - J(G, \overline{\mathcal{M}}(G))] \geq \\ &\alpha - E[J(G, \mathcal{M}(G)) - J(G, \overline{\mathcal{M}}(G))] \geq \quad (\text{By def.}) \end{aligned}$$

$$\alpha - E[\max(0, \frac{\alpha}{2} \frac{|G|}{l+1})] \geq \quad (\text{By Ineq. 1})$$

$$\frac{\alpha}{2}. \quad (\text{By } |G| \leq l+1)$$

Consider the following two equivalent random processes to construct random graphs  $G = (U \cup F, E)$  and  $G' = (U \cup F, E')$ .

- Select  $l$  pairs of nodes from  $U \times F$  uniformly at random without replacement. Add an edge between each selected pair in both  $D$  and  $D'$ . Select one other pair of nodes from  $U \times F$  uniformly at random without replacement, denote it as  $(u, f)$ , and add an edge between  $u$  and  $f$  in  $D'$ .
- Select  $l+1$  pairs of nodes from  $U \times F$  uniformly at random without replacement. Add an edge between each selected pair in both  $D$  and  $D'$ . Select one of the edges in  $D$  uniformly at random, denote it as  $(u, f)$ , and remove it from  $D$ .

Note that,  $G$  is a graph chosen uniformly at random from all graphs on  $U \times F$  with  $l$  edges, and  $G'$  is a graph chosen uniformly at random from all graphs on  $U \times F$  with  $l+1$  edges. Moreover,  $(u, f)$  is both an edge selected uniformly at random from the edges inside  $G'$  and it is an edge selected uniformly at random from the edges that are not in  $G$ . Recall that, by definition, we have

$$\frac{\alpha}{2} \leq E[J(\overline{M}(G'), G')] = E\left[\frac{|\overline{M}(G') \cap G'|}{|\overline{M}(G') \cup G'|}\right]$$

$$\leq E\left[\frac{|\overline{M}(G') \cap G'|}{|G'|}\right] = \frac{E[|\overline{M}(G') \cap G'|]}{|G'|}.$$

Note that, if we select one of the edges of  $G'$  uniformly at random, it exists in  $\overline{M}(G')$  with probability at least  $\frac{E[|\overline{M}(G') \cap G'|]}{|G'|} \geq \frac{\alpha}{2}$ . Hence, we have  $(u, f) \in \overline{M}(G')$  with probability at least  $\frac{\alpha}{2}$ . Let  $S$  be the set of all possible outputs of  $\overline{M}(G')$  where the  $(u, f) \in \overline{M}(G')$ . By the definition of differential privacy we have  $\Pr(\overline{M}(G') \in S) \leq e^\epsilon \Pr(\overline{M}(G) \in S)$ , Which means  $\Pr(\overline{M}(G) \in S) \geq e^{-\epsilon} \Pr(\overline{M}(G') \in S) \geq \frac{\alpha e^{-\epsilon}}{2}$ . This means that  $(u, f) \in \overline{M}(G)$  with probability at least  $\frac{\alpha e^{-\epsilon}}{2}$ . Recall that, by definition  $(u, f)$  is an edge chosen uniformly at random from the edges that do not exist in  $G$ . Hence, if we select one of the edges that do not exist in  $G$ , it exists in  $\overline{M}(G)$  with probability at least  $\frac{\alpha e^{-\epsilon}}{2}$ .

On the other hand, similar to  $G'$ , if we select one of the edges of  $G$  uniformly at random, it exists in  $\overline{M}(G)$  with probability at least  $\frac{\alpha}{2}$ . Hence, we have  $E[|\overline{M}(G)|] \geq (nm - l) \frac{\alpha}{2} e^{-\epsilon} + l \frac{\alpha}{2} \geq \frac{nm\alpha e^{-\epsilon}}{2}$ . Recall that by construction we have  $|\overline{M}(G)| \leq \frac{2(l+1)}{\alpha}$ . This together with the above inequality gives us  $\frac{nm\alpha e^{-\epsilon}}{2} \leq \frac{2(l+1)}{\alpha}$ . This implies  $\epsilon \geq \log \frac{\alpha^2 nm}{4(l+1)} \in \Omega(\log(\alpha^2 nm))$ , as claimed.  $\square$

#### 4 ALGORITHM AND ANALYSIS

In this section we develop an algorithm that find a smooth- $k$ -anonymization of  $G$ . We say an algorithm  $alg$  is  $\alpha$ -approximation if  $J(E, E_{alg})/J(E, E_{Opt}) \geq \alpha$ , where  $E_{alg}$  is the output of  $alg$ ,  $E_{Opt}$  is the optimal solution, and  $J(\cdot, \cdot)$  is the Jaccard similarity function. Our main contribution is captured by the following theorem. We present the proof of this theorem in the full version.

**THEOREM 4.1.** Assume  $J(E, E_{Opt}) \geq 0.75$ . There exists an algorithm that finds a constant approximate smooth- $k$ -anonymization of  $G$  in polynomial time.

At a high level, our algorithm decomposes the users into clusters, each of size at least  $k$ . Then in each cluster  $c$ , for each item  $f$ , if the majority of the vertices in  $c$  have an edge to  $f$ , it adds edges to  $f$  from all nodes in  $c$ ; otherwise it removes the edges to  $f$  from all nodes in  $c$ .

In our algorithm we use metric facility location. In the metric facility location problem we are given a set of points and a set of facilities in a metric space, with an opening cost for each facility. The objective is to select a set of facilities and assign each point to a facility such that the total cost of the selected facilities plus the total distance of the points from their assigned facilities is minimized. Again here, we refer to the set of points assigned to each facility as a cluster. Below is our algorithm Alg. This algorithm depends on a parameter  $\alpha$  which in theory is set to 0.004.

- (1) Embed each user in  $\mathbb{R}^m$  as before.
- (2) For each user  $u_i$  define a facility with the same coordinates and opening cost  $\frac{2\alpha}{1-\alpha} \sum_{u' \in U_i^k} \text{Dist}(u', u_i)$ , where  $U_i^k$  is the set of  $k$  closest points to  $i$ .
- (3) Approximately solve the facility location instance using [7].
- (4) Iteratively, remove each cluster with fewer than  $\alpha k$  points and assign its points to their second closest facility.
- (5) Arbitrarily merge clusters with size less than  $k$  to reach size  $k$ , but do not let the clusters grow larger than  $2k$ .<sup>1</sup>
- (6) For each cluster  $c$ , for each item  $f$ , if most vertices in  $c$  have an edge to  $f$ , add all edges from nodes in  $c$  to  $f$ , otherwise remove all edges from nodes in  $c$  to  $f$ .

#### 5 EXPERIMENTAL RESULTS

We give a brief overview of the empirical performance of our algorithm. We give the full details of the setup as well as additional empirical results in the supplementary material.

**Datasets** Our datasets are as follows: **stochastic** is generated from the stochastic block model; **adult**<sup>2</sup> and **playstore**<sup>3</sup> consist of sparse binary matrices; **dblp** [11], **stanford** [5] consists of adjacency matrices of sparse bipartite graphs and, **user-lists** is a proprietary dataset from a major internet company containing user-interest relationships.

**Baselines and Algorithms** As baselines we use the *Mondrian* anonymization algorithm [4] implementation<sup>4</sup> which enforces  $k$ -anonymity by suppression, as well as the standard randomized response algorithm which enforces differential privacy. We also compare our algorithm for smooth  $k$ -anonymization with an additional baseline *non-smooth* (which uses a simple heuristic to obtain (standard)  $k$ -anonymity by suppression using the clusters obtained by our algorithm). Our algorithm run with  $\alpha = 1/2$ . For the large-scale user-list dataset, we use a simple heuristic (described in the full version) to parallelize our algorithm.

<sup>1</sup>If needed, break a large cluster into some clusters of size at least  $\alpha k$ , so that the total size of small clusters is more than  $k$ . This modification does not change the proof.

<sup>2</sup><https://archive.ics.uci.edu/ml/datasets/adult>

<sup>3</sup><https://www.kaggle.com/lava18/google-play-store-apps>

<sup>4</sup><https://github.com/qiyuangong/Mondrian>

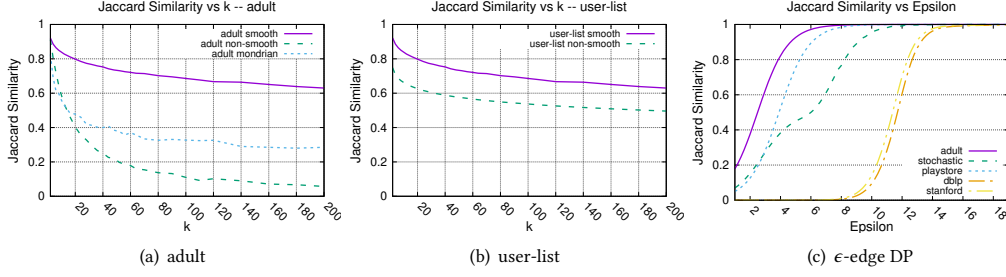


Figure 2: Mean Jaccard similarity for the various datasets and algorithms.

dataset	algorithm	Jaccard	Supp.	Created
adult	mondrian	59.9%	40.1%	0.0%
	non-smooth	64.8%	35.2%	0.0%
	smooth	85.0%	8.9%	7.2%
playstore	mondrian	51.2%	48.8%	0.0%
	non-smooth	39.2%	60.8%	0.0%
	smooth	66.1%	26.2%	11.6%
user_lists	non-smooth	67.3%	32.7%	0.0%
	smooth	71.0%	27.7%	1.9%

Table 1: Average results for  $k = 8$ .

**Jaccard similarity vs  $k$**  First, we evaluate the quality of our algorithm for smooth- $k$ -anonymity for different  $k$  values and we compare it with that of the (non-smooth)  $k$ -anonymity solution and *mondrian*. In Figures 2(a) and 2(b) we show a sample of plots of the mean Jaccard similarity for a given setting of the  $k$  parameter for smooth- $k$ -anonymity (solid line), non-smooth anonymity (dashed line) and *mondrian* (dotted). We were not able to run the *mondrian* algorithm on the larger datasets because, contrary to our algorithm, it scales with the size of the full  $n \times m$  matrix size ( $m$  number of columns) and it does not exploit the sparsity of the matrix. As expected, the Jaccard similarity decreases with increasing  $k$ , but at every  $k$  level smooth- $k$ -anonymity allows to obtain significantly better results than all baselines (in some cases even twice better). We report more detailed results in Table 1 for  $k = 8$ . Notice that our smooth algorithm allows significantly higher jaccard similarity (and lower suppressed entries) for a small increase in created entries. For instance, in adult, the number of suppressed entries is decreased by  $\sim 26\%$  with just a  $\sim 7\%$  increase in added entries.

**Differential privacy** We now evaluate the Jaccard similarity obtained by the  $\epsilon$ -differentially private method. We report the results in Figure 2(c). Here we report results for the lower protection level of  $\epsilon$ -edge differential privacy, as  $\epsilon$ -node differential privacy protection generates results close to random outputs. As expected the sparser the dataset the worse the performance of differential privacy at parity of  $\epsilon$ . Notice how to get Jaccard similarity above 0.5 in stanford or dblp,  $\epsilon$  must be 10 which is too large to provide strong guarantees. We can use these results to compare  $k$ -anonymity and  $\epsilon$ -differential from their a standpoint. We observe that depending on the dataset an anonymity of  $k = 16$  might require an  $\epsilon$  as large as 11 to obtain the same utility.

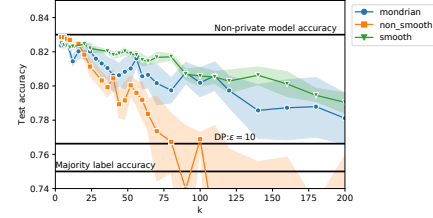


Figure 3: Accuracy in learning task in anonymous data

**Learning from anonymous data** Finally, we report results on using the anonymized datasets in a downstream machine learning task. We use the anonymized version of the adult dataset to learn a classifier for the standard classification task of predicting whether an adult's income is  $\geq \$50k$  per year. The results are reported in Figure 3. Notice our algorithm performs better (or on par) with the best baseline (*mondrian*). We observe (see the supplementary material) that smooth performs significantly better than the  $\epsilon$ -node differentially private algorithm with  $\epsilon = 10$  even for  $k = 200$ , mirroring the degradation seen in the Jaccard similarity metric.

## REFERENCES

- [1] Gagan Aggarwal, Tomas Feder, Krishnaram Kenthapadi, Rajeev Motwani, Rina Panigrahy, Dilys Thomas, and An Zhu. 2005. Approximation algorithms for  $k$ -anonymity. *Journal of Privacy Technology (JOPT)* (2005).
- [2] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially Private Empirical Risk Minimization. *J. Mach. Learn. Res.* (2011).
- [3] Cynthia Dwork and Adam D. Smith. 2010. Differential Privacy for Statistics: What We Know and What We Want to Learn. *J. Priv. Confidentiality* 1, 2 (2010).
- [4] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. 2006. Mondrian multidimensional  $k$ -anonymity. In *ICDE'06*. IEEE, 25–25.
- [5] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. 2009. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics* 6, 1 (2009), 29–123.
- [6] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007.  $t$ -closeness: Privacy beyond  $k$ -anonymity and  $l$ -diversity. In *ICDE*. IEEE, 106–115.
- [7] Shi Li. 2013. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation* 222 (2013), 45–58.
- [8] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramkrishnan Venkatasubramanian. 2007.  $l$ -diversity: Privacy beyond  $k$ -anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 3–es.
- [9] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust De-anonymization of Large Sparse Datasets. In *S&P*. 111–125.
- [10] Latanya Sweeney. 2002.  $k$ -anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.
- [11] Jaewon Yang and Jure Leskovec. 2015. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems* (2015).