Smooth Anonymity for Sparse Graphs

Alessandro Epasto
Google Research

Hossein Esfandiari
Google Research

Vahab Mirrokni
Google Research

Andres Munoz Medina
Google Research

Sergei Vassilvitskii
Google Research

ABSTRACT

When working with user data providing well-defined privacy guarantees is paramount. In this work, we aim to manipulate and share an entire sparse dataset with a third party privately. In fact, differential privacy has emerged as the gold standard of privacy, however, when it comes to sharing sparse datasets, e.g. sparse networks, as one of our main results, 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. In such situations, we need to look into other privacy notions such as k-anonymity. In this work, we consider a variation of k-anonymity, which we call smooth k-anonymity, and design simple large-scale algorithms that efficiently provide smooth k-anonymity. We further perform an empirical evaluation to back our theoretical guarantees and show that our algorithm improves the performance in downstream machine learning tasks on anonymized data.

ACM Reference Format:
Alessandro Epasto, Hossein Esfandiari, Vahab Mirrokni, Andres Munoz Medina, and Sergei Vassilvitskii. 2023. Smooth Anonymity for Sparse Graphs. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In fact, when working with user data, maintaining user privacy is absolutely essential. In this work we study a situation where we intend to share an entire (manipulated) dataset, without violating user privacy. Then 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 the privacy there are a large body of work that attempt to provide formal privacy measures. 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 (typically the presence or absence of any individual user) 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. Both approaches have their own pros and cons, which we briefly discuss next. (We defer the formal definitions to Section 3.)

The main advantage of differential privacy is that the output of a differentially private algorithm remains such even in the face of arbitrary post-processing by an adversary armed with additional side information about the users. This is one reason why it has emerged as the gold standard of privacy. However, as we share more information the differential privacy measure gets weaker. In this work we prove that sharing sparse binary matrices with differential privacy guarantees is infeasible (See Theorem 4.3). Roughly speaking, we prove that any differentially private algorithm either provides a very weak privacy guarantee, or significantly changes the dataset, destroying the underlying signal.

On the other hand, k-anonymity [34] is a popular pre-processing technique that can be used to provide some level of privacy. While k-anonymity can be vulnerable to certain attacks [32], it still provides meaningful guarantees when adversaries have limited access to side information [6]. Moreover, in cases where a data analyst cannot withstand noise, it still represents a formal way to give privacy protections. Making a dataset k-anonymous while best preserving utility is an NP-hard problem [2]. 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 (or anything smaller than a $1 - O\left(\frac{1}{\log(k)}\right)$ fraction). In those cases the algorithm that just returns a null dataset achieves the same guarantee.

In this work, we strive to design an approach for sharing a binary matrix, while respecting the privacy of the users. In order to do this we study a variant of k-anonymity (called smooth-k-anonymity). Then 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. This representation captures the following common setups:
Bipartite Graphs: The nodes of one side correspond to the users and the nodes of the other side corresponds to the features. If user u has feature f, there is an edge between u and f.
User Lists: We have a collection of lists of users, and each list is associated with a feature. If user u has feature f, user u exists in the list associated with f.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Points in a Binary Space: Each user is associated with a point. The coordinates of the point are equivalent to the respective row in the matrix representation.

2 RELATED WORK

The problem of anonymizing data is very well studied. One of the first techniques for anonymizing data sets was k-anonymity [34]. This notion was intended for tabular data where each row corresponds to a user and each column corresponds to a particular feature. The authors define k-anonymity in terms of quasi-identifiers. That is, columns in the data set that, combined, could single out a user. A k-anonymous dataset is one where every user is indistinguishable from k-other users with respect to the quasi-identifier set only (that means that the columns not corresponding to quasi-identifiers are not anonymized). Other works have improved upon this definition by enforcing other restrictions such as requiring l-diversity [25] or t-closeness [22] for non quasi-identifiers, on top of k-anonymity for quasi-identifiers. 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 [28].

The majority of work on k-anonymity has been focused on finding the optimal k-anonymous dataset. That is, one that approximates the original data the best. [26] showed that this task is in fact NP-hard, although it admits a $O(k \log k)$ approximation with running time exponential in k. Later on, [2] obtained a polynomial time approximation of $O(k)$ which was improved by [18, 31] to a $O(\log k)$ approximation. Several variants of these algorithms including using set cover approximations [35]. In addition to these algorithms with provable guarantees, other work has provided heuristics for different notions of anonymization. For instance [19] has defined a heuristic algorithm for k-anonymization of quasi-identifiers based on the construction similar to that of kd-trees [13]. Other authors have defined heuristics based on clustering [7, 37]. None of those methods have provable guarantees in our context.

Another related work to ours is that of [9], which defines the notion of k-isomorphism in social network graphs. Essentially, a graph is k-isomorphic if it can be decomposed into a union of k distinct isomorphic sub-graphs. This notion of anonymity is limited to social network graphs as the goal is to prevent an attacker from identifying a user based on the structure of their neighborhood.

Another framework for achieving anonymity is differential privacy [10]. Unlike k-anonymity, differential privacy provides mathematical guarantees on the amount of information that can be gained by an attacker that observes a differentially private dataset. Differential privacy has been effectively applied for statistics release [11] and empirical risk minimization [8] 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. To release a full dataset in a differentially private manner, [16] introduces the notion of local differential privacy. Local differential privacy allows us to release a full dataset while protecting the information of all users.

Methods from differential privacy have also been used for the release of private graph information. For instance, Nissim et al. [30] shows how to compute the minimum spanning trees and the number of triangles in a graph. Eliáš et al. [12] recently showed how to preserve cuts in graphs with differential privacy.

Kasiviswanathan et al. [17] introduce the notions of edge and node differential privacy in graph settings and show how to calculate functions over graphs under node differential privacy by capping the number of edges per node. Arguably the work most related to this paper is that of Nguyen et al. [29]. The authors propose edge-differential privacy in order to release an anonymous graph. Similar to our results in Corollary 4.2, the authors show that a value of ϵ in $\Omega(\log n)$ is needed in order to achieve non-trivial utility guarantees, where n is the number of nodes in the graph.

We observe that some work has been devoted to combining differential privacy with k-anonymity guarantees. Li et al. [23] show that enforcing k-anonymity in certain data-oblivious ways on a sub-sampled dataset, is sufficient to show differential privacy guarantees.

Our algorithmic techniques are related to the lower bounded facility location problem [3, 14, 33]. This problem has been first introduced and studied independently by Karget and Minkoff [15] and Guha et al. [14]. Lower bounded clustering problems have been motivated by privacy purposes [27]. They both provide bicriteria approximation algorithms for this problem. Later, Svitkina [33] provided a 448-approximation algorithm for this problem. This is the first constant approximation algorithm for this problem. Ahmadian and Swamy [3] improved Svitkina's result and give an 82.6 approximation algorithm for this problem. To the best of our knowledge the latter is the best approximation algorithm for the lower bounded facility location problem.

3 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 $U = \{u_1, \ldots, u_n\}$ denote a set of users and $F = \{f_1, \ldots, f_m\}$ 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 $\mathcal{M} \colon \mathbb{G} \to \mathbb{G}$ 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 3.1. *Edge differential privacy. We say a randomized mechanism $\mathcal{M}$ preserves ϵ-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 $A \subset \mathbb{G}$:*

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

Edge differential privacy implies that the output of a mechanism does not change too much if a single edge of the input graph is changed. Thus an adversary that observes the output of $\mathcal{M}(G)$ may not be able to infer if a single edge was present or not in the graph.

However, if a user has a high degree in G, then the output of an edge-differentially private algorithm may still leak information about the presence or absence of that user in the graph. This leads to a definition of *node-differential privacy* which we detail below.

Definition 3.2. *We say that two graphs $G = (U \cup F, E), G' = (U \cup F, E') \in \mathbb{G}$ are node neighboring if there exists $u \in U$ such that $|E \oplus E'| = |\{e \in E : e = (u, f) \text{ for } f \in F\} \oplus \{e \in E' : e = (u, f) \text{ for } f \in F\}$*

That is, two graphs are node neighboring if one can be obtained from the other by replacing all the edges of a single user.

Definition 3.3 (Node differential privacy)**.** *We say a mechanism $\mathcal{M}$ preserves node differential privacy if for any two node neighboring graphs G and G' and for all $A \subset \mathbb{G}$*

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

Under this notion of anonymity, it is very unlikely for an adversary to identify a user in a particular dataset. While the above notions of differential privacy provide quantifiable protection against an attacker, as we will see, they also require adding a non-trivial amount of noise.

For this reason we revisit an older notion of privacy: k-anonymity. While the original definition of k-anonymity [34] requires defining quasi-identifiers, in this work we assume that every feature can be used as a quasi-identifier.

We first introduce some notation. 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 3.4 (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 $u \in U$, $|C_u(E')| \geq k$.*

The mechanism $\mathcal{M}$ is k-anonymous by suppression if it also satisfies

(1) *$E' \subset E$*

That is the set of items associated with each user in the output graph, is the same of that of at least k users. Moreover, 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 3.5 (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 $u \in U$, $|C_u(E')| \geq k$.*

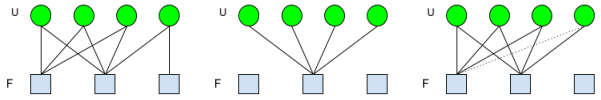


Figure 1: Depiction of k-anonymity with suppression and smooth-k-anonymity for $k = 4$. (left) Original input graph G. (center) k-anonymous with suppression graph. Notice the removal of the edges to the first and last feature. (right) smooth-k-anonymous graph, we preserve the edges to the first feature and add a new edge to it.

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

This definition is very similar to Definition 3.4. 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. 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 3.6. *Given two graphs $G = (U \cup F, E), G' = (U \cup F, E')$ we denote the Jaccard similarity of them by $J(G, G') := \frac{|E \cap E'|}{|E \cup E'|}$.*

4 COMPARISON OF PRIVACY NOTIONS

Here we introduce the new algorithmic problem of finding the best smooth-k-anonymization of a graph. For this reason, in this section, we provide some comparison between smooth-k-anonymity and alternative privacy notions that can be used for data release.

4.1 Comparison with differential privacy

Node differential privacy As we have briefly discussed, node-differential privacy provides the best theoretical guarantees for privacy protection. In the specifics of our setup, node-differential privacy is equivalent to the so-called *local* differential privacy [16], where every user is acting separately without coordination from some global authority.

Let $G = (U \cup F, E)$, borrowing from local differential privacy ideas, one way of achieving node differential privacy is by releasing $\mathcal{M}(G) = (U \cup F, E')$ built according to Algorithm 1. The algorithm is parameterized by a randomized response probability p. It is not hard to show that in order to achieve ϵ-node differential privacy $p = \frac{2}{1+e^{\frac{\epsilon}{|F|}}}$. Notice that this value converges to 1 exponentially fast as a function of the size of the feature set F. That is, even for relatively small graphs, in order to achieve any meaningful privacy guarantee, the probability of returning a completely random graph is very close to 1. For this reason, this notion is not amenable to be used with good utility in our setting.

Edge differential privacy Algorithm 1 can also be used to define a mechanism $\mathcal{M}$ that is edge differential privacy. In that case

Algorithm 1 Randomized response

Input: $G = (U \cup F, E)$, randomized response prob. p
Output: anonymized graph $G' = (U \cup F, E')$.
for $u \in U, f \in F$
  Sample $Y \sim$ Bernoulli(p)
  if $Y = 1$ **then** $(u, f) \in E'$ with probability 1/2.
  else if $(u, f) \in E$ **then** $(u, f) \in E'$

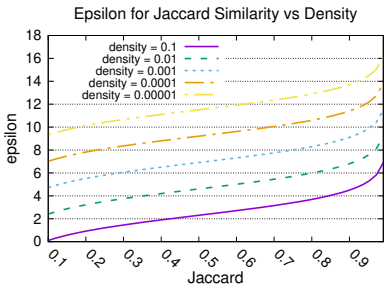


Figure 2: The ϵ necessary for a given Jaccard, as a function of density

.

one can achieve ϵ-edge differential privacy by setting $p = \frac{2}{1+e^{\epsilon}}$ (see the supplementary material).

In this section fist we consider Algorithm 1 as a natural way to provide differential privacy and upper bound the Jaccard similarity of the input and the output graphs of this algorithm. Later in this section we show that a similar bound holds for all differential privacy algorithms.

We now upper bound the similarity of the output of the algorithm with ϵ-edge differential privacy. The bound depends on the density λ of the graph G defined as: $\lambda(G) = \frac{|E|}{|U||F|}$. The proof of this theorem is available in the supplementary material.

Theorem 4.1. *Let $G = (U \cup F, E) \in \mathbb{G}$ be a graph, $\delta > 0$. Let $\mathcal{M}$ be a mechanism that generates a graph according to Algorithm 1 with $p = \frac{2}{1+e^{\epsilon}}$. Let $Q = |U||F|$ and $C(\delta) = \frac{\log(2/\delta)}{2}$. If $\frac{p}{4} \geq \sqrt{\frac{C(\delta)}{Q}}$, then with probability at least $1 - \delta$:*

$$J(\mathcal{M}(G), G) \leq \frac{1 - \frac{1}{e^{\epsilon}+1}}{1 + \frac{1}{e^{\epsilon}+1}\frac{1-\lambda}{\lambda}} + 2\sqrt{\frac{C(\delta)}{Q}},$$

where $\lambda := \lambda(G)$.

Using Theorem 4.1, we can plot in Figure 2 a lower bound for the ϵ needed to achieve a certain level of Jaccard similarity utility, given a density factor λ. Notice that for reasonably sparse datasets, say graphs with density around 1/10,000, one needs an ϵ higher than 10 to obtain a Jaccard Similarity of more than 50%. This result will be confirmed in Section 6 in our empirical analysis.

The previous Theorem allows us to derive the following.

Corollary 4.2. *If the average degree is $O(m^{0.99})$, for any $\epsilon \in o(\log m)$, ϵ-DP gives a solution with Jaccard similarity of $o(1)$.*

This means that, when the average user-degree is poly-logarithmic (or even $m^{0.99}$) we need $\epsilon \in \Omega(\log m)$ to achieve a constant Jaccard similarity with ϵ-edge differential privacy. It is well-known that many real-world datasets are sparse. For instance in the context of real-world networks, classical theoretical models [4] as well as empirical studies [5, 20] postulate constant average degree or degrees growing slower than the size of the graph.

The above results show that using randomized response, any differentially private approximation to a graph with high utility requires an exceptionally large value of ϵ, thus rendering void any privacy guarantees. Later in Theorem 4.3 we show a similar bound on ϵ for all differentially private algorithms. In fact, for such a high ϵ, the algorithm likely maintains the graph G unmodified thus exposing users to re-identification risks. For this reason we believe k-anonymity might in fact provide better protection in practice, especially in scenarios where our goal is to produce a private version of the input graph with high utility.

Unfortunately, however, all of the previous work [2, 18] provide non-trivial guarantees on the quality of a k-anonymous mechanism only when $J(E, E_{\text{Opt}}) \geq 1 - O(\frac{1}{\log k})$, in other words when very few edges need to be removed, as the similarity between the graph and the optimal k-anonymous graph is very high. By contrast, we show in Section 5 one can achieve a constant approximation to the optimal smooth-k-anonymous solution in less restrictive scenarios.

Hardness of Differential Privacy So far in this section we analyzed randomized response and show that this mechanism requires an unacceptably large ϵ. One may ask if there is any other ϵ-differential privacy mechanism with a small ϵ that guarantees the output to be similar to the input. The following theorem rules out the existence of such a mechanism.

Theorem 4.3. *Let $\mathcal{M}$ be an arbitrary mechanism that satisfies ϵ-edge differential privacy. Let α 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 $\epsilon \in \Omega\big(\log(\alpha^2 nm)\big)$.*

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 ϵ-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 ϵ. We define policy $\overline{\mathcal{M}}(G)$ 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 ϵ-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

$$E[J(G, \overline{\mathcal{M}}(G))] = E[J(G, \mathcal{M}(G))] - E[J(G, \mathcal{M}(G)) - J(G, \overline{\mathcal{M}}(G))] \geq$$
$$\alpha - E[J(G, \mathcal{M}(G)) - J(G, \overline{\mathcal{M}}(G))] \geq \quad \textit{(By def.)}$$

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

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

$$\frac{\alpha}{2} \leq E\Big[J(\overline{\mathcal{M}}(G'), G')\Big] = E\Big[\frac{|\overline{\mathcal{M}}(G') \cap G'|}{|\overline{\mathcal{M}}(G') \cup G'|}\Big]$$
$$\leq E\Big[\frac{|\overline{\mathcal{M}}(G') \cap G'|}{|G'|}\Big] = \frac{E\big[|\overline{\mathcal{M}}(G') \cap G'|\big]}{|G'|}.$$

Note that, if we select one of the edges of G' uniformly at random, it exists in $\overline{\mathcal{M}}(G')$ with probability at least $\frac{E\left[|\overline{\mathcal{M}}(G') \cap G'|\right]}{|G'|} \geq \frac{\alpha}{2}$. Hence, we have $(u, f) \in \overline{\mathcal{M}}(G')$ with probability at least $\frac{\alpha}{2}$. Let S be the set of all possible outputs of $\overline{\mathcal{M}}(G')$ where the $(u, f) \in \overline{\mathcal{M}}(G')$. By the definition of differential privacy we have

$$\mathbf{Pr}\Big(\overline{\mathcal{M}}(G') \in S\Big) \leq e^{\epsilon}\mathbf{Pr}\Big(\overline{\mathcal{M}}(G) \in S\Big),$$

Which means

$$\mathbf{Pr}\Big(\overline{\mathcal{M}}(G) \in S\Big) \geq e^{-\epsilon}\mathbf{Pr}\Big(\overline{\mathcal{M}}(G') \in S\Big) \geq \frac{\alpha e^{-\epsilon}}{2}.$$

This means that $(u, f) \in \overline{\mathcal{M}}(G)$ 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 $\overline{\mathcal{M}}(G)$ 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 $\overline{\mathcal{M}}(G)$ with probability at least $\frac{\alpha}{2}$. Hence, we have

$$E[|\overline{\mathcal{M}}(G)|] \geq (nm - l)\frac{\alpha}{2}e^{-\epsilon} + l\frac{\alpha}{2} \geq \frac{nm\alpha e^{-\epsilon}}{2}.$$

Recall that by construction we have $|\overline{\mathcal{M}}(G)| \leq \frac{2(l+1)}{\alpha}$. This together with the above inequality gives us $\frac{nm\alpha e^{-\epsilon}}{2} \leq \frac{2(l+1)}{\alpha}$. This implies $\epsilon \geq \log \frac{\alpha^2 nm}{4(l+1)} \in \Omega\big(\log(\alpha^2 nm)\big)$, as claimed. □

4.2 Comparison with k-anonymity by suppression

In this section we compare k-anonymity by suppression with smooth-k-anonymity. First, in terms of privacy, we notice that both smooth-k-anonymity and k-anonymity by suppression guarantee that every user in the output is indistinguishable from at least k-users. Moreover, observe that since smooth-k-anonymity is allowed to add edges to the output graph, an attacker would not be certain whether an edge was in the original graph or not.

We now show formally that the optimum solution of smooth-k-anonymity may preserve a significantly larger fraction of the input data than regular k-anonymity. To show this separation rigorously, we adopt the bipartite stochastic block model (SBM), which is commonly used in modeling applications, for instance in clustering and community detection [1].

Bipartite Stochastic Block Model. To define the bipartite SBM, consider the following random process. We have two sets of n vertices, and each set is further decomposed into r blocks of size s, where $r \cdot s = n$. Each block in the first part corresponds to one block of the second part. There is an edge between each pair of vertices in two corresponding blocks independently with probability q, and between every other pair of vertices with probability p. We let $\alpha = qs$ denote the expected number of edges that one node has to its corresponding block, a.k.a. *internal edges*. We let $\beta = p(n - s)$ denote the expected number of edges that one node has to vertices other than its corresponding block, a.k.a. *external edges*. We refer to this as the stochastic block model with parameters r, s, α, β.

The result that we provide in this section is of particular interest when the blocks are not very sparse, i.e., $\alpha \in \omega(\frac{\log n}{\log 1/q})$ and $\alpha \in \Omega(\beta + s)$.

The next theorem upper bounds the number of edges in a (non-smooth) k-anonymous subgraph of a graph generated by the stochastic block model by $O(n\frac{\log n}{\log 1/q})$. Therefore, since $\alpha \in \omega(\frac{\log n}{\log 1/q})$, the fraction of remaining edges tends to zero. The proof of this theorem is presented in the supplementary material.

Theorem 4.4. *Let G be a graph generated by the stochastic block model with parameters r, s, α, β. Let $k \geq \frac{2\log n}{\log 1/q}$. With probability 99%, any k-anonymous subgraph of G contains at most $\frac{2\log n+10}{\log 1/q}n \in \tilde{O}(n)$ edges.*

This allows us to show a gap with smooth-k-anonymization. In fact, a natural solution that puts the vertices of each block in a cluster leads to a solution for smooth-k-anonymity that in expectation keeps αn edges, adds $(s - \alpha)n$ edges, and removes βn edges. Since $\alpha \in \Omega(\beta + s)$, the number of remaining edges αn is not less than a constant factor of the changed edges. This result concerns the optimum solution, but in the next section we provide an algorithm for computing smooth-k-anonymization of a graph.

5 ALGORITHMS AND ANALYSIS

In this section we develop algorithms that find a smooth-k-anonymization of G. We say an algorithm *alg* is α-approximation if $J(E, E_{alg})/J(E, E_{\text{Opt}}) \geq \alpha$, 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.

Theorem 5.1. *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.

5.1 Preliminaries

Let us start with some preliminary notions and lemmas. We will abuse notation slightly, and for a user u and item f, say $u \in f$ if there is a (u, f) edge in the graph.

Note that we can represent each user u with a point in a m dimensional space. For an item f_i, we set the i-th dimension of u's representation to 1 if $u \in f_i$ and to 0 otherwise. In this space we define the distance of two points, u and v to be the number of positions where u and v differ (*i.e.* their Hamming distance).

Let Δ_u^{Alg} be the number of positions that the algorithm Alg changes in the binary vector corresponding to the user u. Intuitively, $\sum_u \Delta_u^{\text{Opt}}$ should be related to $J(E, E_{\text{Opt}})$. The following lemma formalizes this intuition.

Lemma 5.2. *Assume $J(E, E_{Opt}) \geq 1 - \phi$. We have $\sum_u \Delta_u^{Opt} \leq \frac{2\phi}{1-\phi}|E|$.*

All proofs from this section are in the supplementary material.

To complement Lemma 5.2, the following lemma lower bounds $J(E, E_{\text{Alg}})$ given an upper bound on $\sum_u \Delta_u^{\text{Alg}}$.

Lemma 5.3. *Assume $\sum_u \Delta_u^{Alg} \leq \phi'|E|$. We have $J(E, E_{Alg}) \geq 1 - \frac{\phi'}{2}$.*

5.2 Initial algorithm

We now provide an approximation algorithm using a reduction to *lower-bounded r-median*[1]. In the next subsection we improve it using a slightly more complicated algorithm.

In the lower-bounded r-median problem we are asked to select at most r centers from n points and assign each point to one center such that (i) the number of points assigned to each center is at least k, (ii) the total distance of the points from their assigned centers is minimized. We refer to each set of the points that are assigned to the same center as a cluster. In this paper we let $r = n/k$, which means that the algorithm may use as many centers as it needs, however, it must assign at least k points to each center[2]. Here we use a 82.6 approximation algorithm for lower-bounded r-median [3], which is the best known result to the best of our knowledge. We refer to this algorithm as Alg_1.

(1) Embed each user in $\mathbb{R}^m$ as described at the beginning of this section.
(2) Approximately solve the lower-bounded r-median on the points (for $r = n/k$).
(3) 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.

Note that by definition of lower-bounded r-median, each cluster contains at least k points. Moreover, the data that we output for users that belong to the same cluster are the same. Hence, the output satisfies the anonymity part of the smooth-k-anonymity condition. Moreover, the output satisfies the majority part of the smooth-k-anonymity assumptions. Next we bound $J(E, E_{\text{Alg}_1})$ assuming $J(E, E_{\text{Opt}}) \geq 1 - \phi$.

For analysis sake, we introduce the *relaxed lower-bounded r-median* problem in which we are allowed to select any possible discrete point in the space as a center (as opposed to being restricted to select centers only from the points that appear in the input). Note that, if we take a solution to relaxed lower-bounded r-median and move each center to its closest point (that appears in the input), by triangle inequality the cost of the solution increases by at most a factor 2. Therefore the cost of lower-bounded r-median is at most twice that of relaxed lower-bounded r-median.

By Lemma 5.2 we have $\sum_u \Delta_u^{\text{Opt}} \leq \frac{2\phi}{1-\phi}|E|$. We now prove there exists a solution to relaxed lower-bounded r-median with cost at most $\frac{2\phi}{1-\phi}|E|$. Take an optimal anonymous solution and consider the equivalence classes of nodes with the same neighborhood. This induces a clustering of the nodes with clusters of size at least k. Now, observe that the total number of entries changed is equal to the sum of distances from the output neighborhood (of each class) and the original nodes. So this shows that there exists a clustering with sizes at least k with total cost $\sum_u \Delta_u^{\text{Opt}}$.

Therefore, there exists a solution to lower-bounded r-median with cost at most $\frac{4\phi}{1-\phi}|E|$. Note that we are using an 82.6-approximation algorithm to find lower-bounded r-median. Hence, the total cost of our solution is at most $\frac{330.4\phi}{1-\phi}|E|$. The last line of the algorithm does not increase the total cost (since it selects the best center for each cluster). Hence we have $\sum_u \Delta_u^{\text{Alg}_1} \leq \frac{330.4\phi}{1-\phi}|E|$. By applying this to Lemma 5.3 we have $J(E, E_{\text{Alg}_1}) \geq 1 - \frac{165.2\phi}{1-\phi}$. This is a positive constant for any $\phi \leq 0.006$. This implies the following theorem.

Theorem 5.4. *Assume $J(E, E_{Opt}) \geq 0.994$. There exists an algorithm that finds a constant approximation smooth-k-anonymization of G in polynomial time.*

Of course, having $J(E, E_{\text{Opt}}) \geq 0.994$ is a very strong assumption. Next we substantially relax this requirement.

5.3 Improved algorithm

To prove a better algorithm we will use the 1.488 approximation algorithm for the metric facility location problem [24] as a subroutine. 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.

[1] We use r-median instead of k-median to avoid the confusion with the parameter k in k-anonymity. We will use r as the upper bound on the number of clusters and k as the lower bound for the size of the clusters.
[2] This means there are at most n/k centers.

Below is our algorithm Alg_2. This algorithm depends on a parameter α that we set later.

(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.
(4) Iteratively, remove each cluster with fewer than αk 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 $2k$. [3]
(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.

Theorem 5.5. *Assume $J(E, E_{Opt}) \geq 0.75$. Algorithm Alg_2 run with an $\alpha \in O(1)$ finds a constant approximation to smooth-k-anonymization of E in polynomial time.*

Proof. Svitkina [33] showed that Lines 2 to 5 gives solution in which (i) the size of each cluster is at least αk, and (ii) the cost of the solution is at most $1.488 \cdot \frac{1+\alpha}{1-\alpha}$ times that of lower-bounded r-median. Similar to our previous algorithm, since the size of each cluster is at least k, the last line of the algorithm guarantees smooth-k-anonymity. Next we bound $J(E, E_{\text{Alg}_2})$ assuming $J(E, E_{\text{Opt}}) \leq 1-\phi$. We refer to the first four lines of algorithm Alg_2 as Alg_2'. Similar to the previous subsection we know that there exists a solution to lower-bounded r-median with cost at most $\frac{4\phi}{1-\phi}|E|$. Therefore the cost of the solution to the facility location problem is at most

$$5.952 \cdot \frac{1+\alpha}{1-\alpha} \cdot \frac{\phi}{1-\phi}|E|.$$

Again similar to the previous subsection and by applying Lemma 5.3 we have

$$J(E, E_{\text{Alg}_2'}) \geq 1 - 2.976 \cdot \frac{1+\alpha}{1-\alpha} \cdot \frac{\phi}{1-\phi}.$$

Later we show that Line 5 decreases the Jaccard similarity by at most a factor $\frac{\alpha^2}{8}$. Hence we have

$$J(E, E_{\text{Alg}_2}) \geq \frac{\alpha^2}{8}\left(1 - 2.976 \cdot \frac{1+\alpha}{1-\alpha} \cdot \frac{\phi}{1-\phi}\right),$$

which is a positive constant for $\phi \leq 0.25$ and $\alpha = 0.004$. This implies Theorem 5.5.

New we show that Line 5 decreases the Jaccard similarity by a factor of at least $\frac{\alpha^2}{8}$. To prove this, we use the probabilistic method. For each cluster we show a random point such that if we move the points in each cluster to its corresponding random point, the expected Jaccard similarity is at least $\frac{\alpha^2}{8}$ times that of the initial Jaccard similarity. This implies that there exists a fixed (i.e., deterministic) set of points such that if we move the points in each cluster to its corresponding fixed point, the expected Jaccard similarity is at least $\frac{\alpha^2}{8}$ times that of the initial Jaccard similarity. Note that, for each cluster, we are selecting the optimum center, and hence this statement holds for our selected centers as well.

[3] If needed, break a large cluster into some clusters of size at least αk, so that the total size of small clusters is more than k. This modification does not change the proof.

Let E_{init} be the edge set corresponding to the clustering prior to Line 5. For each merged cluster we select the center of one of its initial clusters uniformly at random. Note that each merged cluster contains at most $2/\alpha$ initial clusters. we select the center of each initial cluster with probability at least $\alpha/2$. This means that each edge that exists in $E \cap E_{\text{init}}$ exists after merging the clusters with probability at least $\alpha/2$. Hence, the expected number edges that exists after merging is at least $\frac{\alpha}{2}|E \cap E_{\text{init}}|$.

Moreover, the number of nodes in each initial cluster is at most $\frac{\alpha k}{2k}$ fraction of that of the merged cluster. Hence, the total number of edges increases by a factor of at most $2/\alpha$ after merging and moving the point to the random centers. Hence, the expected Jaccard similarity after the merge is at least

$$\frac{\frac{\alpha}{2}|E \cap E_{\text{init}}|}{\frac{2}{\alpha}|E_{\text{init}}| + |E|} \geq \frac{\frac{\alpha}{2}|E \cap E_{\text{init}}|}{2\frac{2}{\alpha}|E \cup E_{\text{init}}|} = \frac{\alpha^2}{8}\frac{|E \cap E_{\text{init}}|}{|E \cup E_{\text{init}}|},$$

as claimed. This completes the proof of Theorem 5.5. □

6 EXPERIMENTAL RESULTS

We give a brief overview of the empirical performance of our algorithms. We give the full details of the setup as well as additional empirical results in the supplementary material. We will release an *open-source* version of our code by the camera-ready deadline.

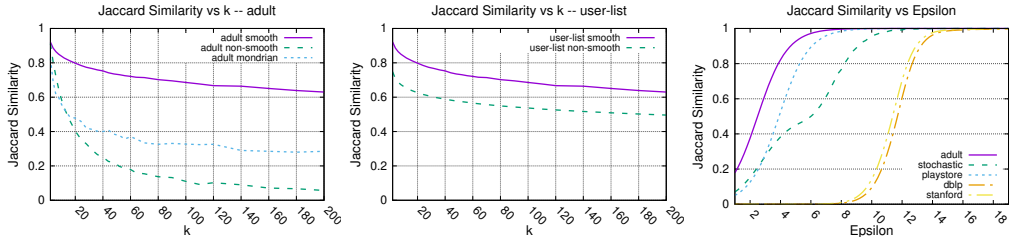
Datasets We used representative examples of sparse binary matrices (and bipartite graphs) of different origins and structural properties. The scales of the datasets are up to $> 1B$ rows, $> 100k$ columns, $> 10B$ entries (for our largest dataset) and the densities of the matrices range from 10^{-5} to 0.1. We used one synthetic dataset, four publicly-available real-world datasets as well as one large-scale proprietary dataset from a major internet company. These are as follows: **stochastic** is generated from the stochastic block model described in Section 4.2; **adult**[4] and **playstore**[5] consist of sparse binary matrices; **dblp** [36], **stanford** [21] consists of adjacency matrices of sparse bipartite graphs and, **user-lists** is a proprietary dataset from a major internet company containing user-interest relationships.

Metrics We measure utility using the *Jaccard similarity* between the set of the edges as defined in the paper as well as the number of *suppressed entries* and *created entries*.

Experimental infrastructure Our algorithm is implemented as a single-threaded C++ problem and is run on standard commodity hardware, with the exception of runs on the large-scale user-list dataset. For this dataset, we evaluated a simple heuristic to parallelize our algorithm. (See the supplementary material for more details.)

Baselines and Algorithms As baselines we use the *Mondrian* anonymization algorithm [19] implementation[6] which enforces k-anonymity by suppression, as well as the randomized response algorithm of Section 4 which enforces ϵ edge- or node-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 is always run with

[4] https://archive.ics.uci.edu/ml/datasets/adult
[5] https://www.kaggle.com/lava18/google-play-store-apps
[6] https://github.com/qiyuangong/Mondrian



(a) adult (b) user-list (c) ϵ-edge DP

Figure 3: 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 $k = 8$ for various algorithms and dataset. Jac., S.E. and C.E. stand for, respectively, Jaccard similarity and fraction of suppressed entries and newly created entries (both normalized by the entries in the input dataset).

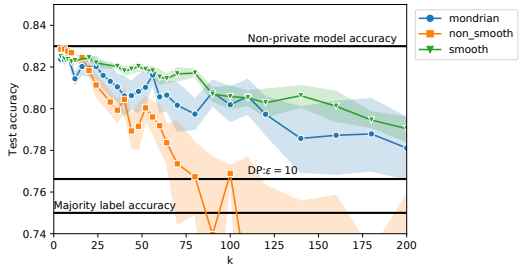


Figure 4: Accuracy in learning task in anonymous data . We also include a baseline of using only the majority label as well as training a model without anonymity and a model that uses node differential privacy with $\epsilon = 10$

$\alpha = 1/2$, which in practice gives good results for all datasets (we observe that values of α in $[1/8, 1/2]$ have similar results).

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 3(a) and 3(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 ϵ-differentially private method. We report the results in Figure 3(c). Here we report results for the lower protection level of ϵ-edge differential privacy, as ϵ-node differential privacy protection generates results close to random outputs. As expected (see Section 4) the sparser the dataset the worse the performance of differential privacy at parity of ϵ. Notice how to get Jaccard similarity above 0.5 in stanford or dblp, ϵ must be 10 which is too large to provide strong guarantees. We can use these results to compare k-anonymity and ϵ-differential from their a standpoint. We observe that depending on the dataset an anonymity of $k = 16$ might require an ϵ as large as 11 to obtain the same utility.

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 4. 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 ϵ-node differentially private algorithm with $\epsilon = 10$ even for $k = 200$, mirroring the degradation seen in the Jaccard similarity metric.

7 CONCLUSION

We presented a new notion of anonymity which relaxes standard k-anonymity and allows us to obtain better guarantees and improved results in downstream machine learning tasks. Our algorithms reduce the anonymization problem to clustering with lower bounds and require non-trivial analysis to prove approximation guarantees. Many interesting questions remain, including strengthening our bounds for smooth k-anonymity and better understanding the interplay between the various notions of privacy.

REFERENCES

[1] Emmanuel Abbe. 2017. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research* 18, 1 (2017), 6446–6531.

[2] 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).

[3] Sara Ahmadian and Chaitanya Swamy. 2012. Improved approximation guarantees for lower-bounded facility location. In *International Workshop on Approximation and Online Algorithms*. Springer, 257–271.

[4] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.

[5] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. 2012. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference*. 33–42.

[6] Raef Bassily, Adam Groce, Jonathan Katz, and Adam D. Smith. 2013. Coupled-Worlds Privacy: Exploiting Adversarial Uncertainty in Statistical Data Privacy. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*. 439–448.

[7] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. 2007. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications*. Springer, 188–200.

[8] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. 2011. Differentially Private Empirical Risk Minimization. *J. Mach. Learn. Res.* (2011).

[9] James Cheng, Ada Wai-Chee Fu, and Jia Liu. 2010. K-isomorphism: privacy preserving network publication against structural attacks. In *SIGMOD*. 459–470.

[10] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

[11] 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).

[12] Marek Eliáš, Michael Kapralov, Janardhan Kulkarni, and Yin Tat Lee. 2020. Differentially Private Release of Synthetic Graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 560–578.

[13] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* 3, 3 (1977), 209–226.

[14] Sudipto Guha, Adam Meyerson, and Kamesh Munagala. 2000. Hierarchical placement and network design problems. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE, 603–612.

[15] DR Karget and Maria Minkoff. 2000. Building Steiner trees with incomplete global knowledge. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE, 613–623.

[16] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2011. What Can We Learn Privately? *SIAM J. Comput.* 40, 3 (2011), 793–826.

[17] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2013. Analyzing Graphs with Node Differential Privacy. In *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013, Tokyo, Japan, March 3-6, 2013. Proceedings*. 457–476.

[18] Batya Kenig and Tamir Tassa. 2012. A practical approximation algorithm for optimal k-anonymity. *Data Mining and Knowledge Discovery* 25, 1 (2012), 134–168.

[19] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. 2006. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*. IEEE, 25–25.

[20] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2007. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 2–es.

[21] 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.

[22] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 106–115.

[23] Ninghui Li, Wahbeh H Qardaji, and Dong Su. 2011. Provably private data anonymization: Or, k-anonymity meets differential privacy. *CoRR, abs/1101.2604* 49 (2011), 55.

[24] Shi Li. 2013. A 1.488 approximation algorithm for the uncapacitated facility location problem. *Information and Computation* 222 (2013), 45–58.

[25] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1, 1 (2007), 3–es.

[26] Adam Meyerson and Ryan Williams. 2004. On the Complexity of Optimal K-Anonymity. In *SIGMOD*. 223–228.

[27] Rajeev Motwani and Shubha U Nabar. 2008. Anonymizing unstructured data. *arXiv preprint arXiv:0810.5582* (2008).

[28] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust De-anonymization of Large Sparse Datasets. In *S&P*. 111–125.

[29] Hiep H. Nguyen, Abdessamad Imine, and Michaël Rusinowitch. 2016. Network Structure Release under Differential Privacy. *Transactions on Data Privacy* 9, 3 (2016), 215–241.

[30] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*. 75–84.

[31] Hyoungmin Park and Kyuseok Shim. 2010. Approximate algorithms with generalizing attribute values for k-anonymity. *Information Systems* 35, 8 (2010), 933–955.

[32] Ganta Srivatsava Ranjit, Kasiviswanathan Shiva Prasad, and Smith Adam. 2008. Composition Attacks and Auxiliary Information in Data Privacy. In *KDD (KDD '08)*. ACM.

[33] Zoya Svitkina. 2010. Lower-bounded facility location. *ACM Transactions on Algorithms (TALG)* 6, 4 (2010), 69.

[34] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.

[35] Shyue-Liang Wang, Yu-Chuan Tsai, Hung-Yu Kao, and Tzung-Pei Hong. 2010. Anonymizing set-valued social data. In *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. IEEE, 809–812.

[36] Jaewon Yang and Jure Leskovec. 2015. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems* (2015).

[37] Wantong Zheng, Zhongyue Wang, Tongtong Lv, Yong Ma, and Chunfu Jia. 2018. K-Anonymity Algorithm based on Improved Clustering. In *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 462–476.