What Should We Protect? Defining Differential Privacy for Social Network Analysis

Christine Task and Chris Clifton

Abstract Privacy of social network data is a growing concern that threatens to limit access to this valuable data source. Analysis of the graph structure of social networks can provide valuable information for revenue generation and social science research, but unfortunately, ensuring this analysis does not violate individual privacy is difficult. Simply anonymizing graphs or even releasing only aggregate results of analysis may not provide sufficient protection. Differential privacy is an alternative privacy model, popular in data-mining over tabular data, that uses noise to obscure individuals' contributions to aggregate results and offers a very strong mathematical guarantee that individuals' presence in the data-set is hidden. Analyses that were previously vulnerable to identification of individuals and extraction of private data may be safely released under differential-privacy guarantees. We review two existing standards for adapting differential privacy to network data and analyze the feasibility of several common social-network analysis techniques under these standards. Additionally, we propose *out-link privacy* and *partition privacy*, novel standards for differential privacy over network data, and introduce powerful private algorithms for common network analysis techniques that were infeasible to privatize under previous differential privacy standards.

1 Introduction

Social networks are powerful abstractions of individuals and the relationships that connect them; social network analysis can be a very powerful tool. For example, understanding how well-connected a network is can aid in the development of

C. Task (✉) · C. Clifton
Department of Computer Science, Purdue University, West Lafayette, IN, USA
e-mail: ctask@purdue.edu

C. Clifton
e-mail: clifton@purdue.edu

F. Can et al. (eds.), *State of the Art Applications of Social Network Analysis*, Lecture Notes in Social Networks, DOI: 10.1007/978-3-319-05912-9_7,

word-of-mouth marketing campaign: How quickly will word of a product spread? Similar analysis is useful in epidemiology, predicting spread of a disease.

However, data about people and their relationships is potentially sensitive and must be treated with care to preserve privacy. Generally, social network graphs are anonymized before being made available for analysis. However, as several recent incidents have demonstrated, releasing even anonymized graphs may lead to re-identification of individuals within the network and disclosure of confidential information, with serious consequences for those involved. In 2007, Netflix released the Netflix Prize data-set, containing anonymized data about the viewing habits of its members, for public analysis by information retrieval researchers. Within a year, it had been demonstrated that wide-spread de-anonymization of individuals in the data-set was possible using public information from the Internet Movie Database [1]. By 2009, Netflix was involved in a lawsuit with one of its members who had been victimized by the resulting privacy invasion.

Privacy researchers have attempted to improve the security provided by graph anonymization techniques by adding noise to the node parameters and structure of the graph [2]. However, even a noisy graph structure with no node parameters whatsoever can be subject to deanonymization, particularly if an attacker has background knowledge of the network data [3]. For example, knowing the friendship relationships of a few individuals can make them identifiable in the released graph, leading to identification of their friends (and disclosure of information, such as other relationships, that those friends might not want publicly revealed.) As global social networks become more broadly accessible, these types of background knowledge are more readily available [3].

Differential privacy is a privacy standard developed for use on tabular data that provides strong guarantees of privacy without making assumptions about an attacker's background knowledge [4]. Differentially-private queries inject randomized noise into query results to hide the impact of adding or removing an arbitrary individual from the data-set. Thus, an attacker with an arbitrarily high level of background knowledge cannot, with a high degree of probability, glean any new knowledge about individuals from differentially-privatized results; in fact, the attacker cannot guess whether any given individual is present in the data at all.

While many of the privacy concerns associated with social-network analysis could be relieved by applying differential-privacy guarantees to common social-network analysis techniques, researchers have struggled to develop suitable adaptations of these techniques. Two principal difficulties arise: The adaptation of differential privacy from tabular data to network data, and the high sensitivity of social-network metrics to relatively small changes in the network structure.

Two models for applying differential privacy to social networks have arisen. *Node privacy* limits the ability of an attacker to learn any information about an individual, but at a high cost in added noise. *Edge privacy* protects against learning any particular relationship, but may allow learning general information about an individual. This chapter introduces *out-link privacy* and *partition privacy*, models for differential privacy that provide greater protection than edge privacy while allowing important

types of analysis that are not feasible under node privacy. The key contributions and outline of this chapter are as follows:

- A straightforward introduction to traditional differential privacy;
- A discussion of two known differential-privacy standards for network data, as well as the contribution of two new standards, *out-link privacy* and *partition privacy*, that provide strong privacy guarantees with the introduction of very small noise;
- A study of the feasibility of common social-network analysis techniques under differential-privacy;
- The contribution of two new algorithms satisfying *out-link privacy* that use ego-network style analysis to provide approximate results for queries that are too sensitive to perform under previous standards.
- A demonstration of the application *partition privacy* to a variety of contexts; *partition privacy* is a new approach that provides unprecedented levels of privacy with minimal noise, for studies that compare variables across multiple social networks. It allows the wide variety of techniques developed for traditional differential privacy to be applied to social-network privacy.

2 Traditional Differential Privacy

Differential privacy was developed by Cynthia Dwork and collaborators at Microsoft Research Labs [4]. It does not define a specific technique or algorithm; instead it states a mathematical guarantee of privacy that sufficiently well-privatized queries can satisfy. Consider a common sequence of events in social science research: a survey is distributed to individuals within a population; a subset of the population chooses to participate in the survey; individual information from the surveys is compiled into a data-set and some analysis is computed over it; the analysis may be privatized by the injection of random noise; and the final privatized result is released to the general public. Differentially-private queries offer a rigorous mathematical guarantee to survey participants that the released results will not reveal their participation in the survey.

We first introduce a few useful notations: I is set of individuals who contribute information to the data-set D_I (e.g., survey participants). The set of *all possible* data-sets is $\mathcal{D}$. We use $F : \mathcal{D} \rightarrow \Re^k$: to refer to the desired non-privatized analysis performed on a data-set and $Q : \mathcal{D} \rightarrow \Re^k$ to refer to the privatized implementation of F. We refer to the publicly released, privatized analysis results as R.

If R are the privatized query results that are released to the public, then R is the only evidence an attacker has about the nature of D_I. We introduce a possible-worlds model to understand how differential privacy works (see Fig. 1). We define D_I to be *true world* from which the analysis was taken. We also define any data-set that differs by the presence or absence of one individual to be a "neighboring" possible world: thus D_{I-Bob} is the neighboring possible world of D_I in which Bob chose to not participate in the survey.

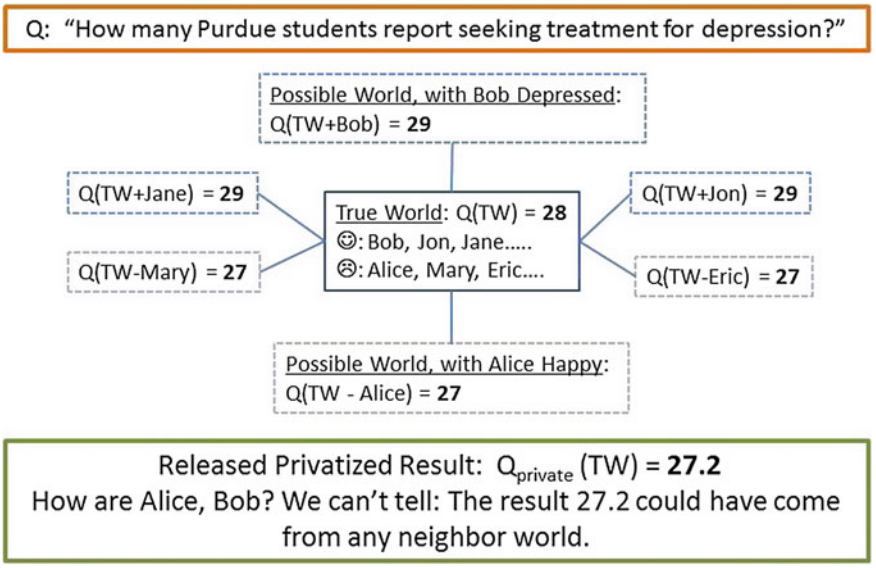


Fig. 1 Differential privacy adds noise to obfuscate individuals' affect on query results

We require that an attacker possessing the privatized results R be unable to determine whether or not Bob (or any other specific individual) took the survey, i.e., whether or not R are the results from an analysis of D_I or D_{I-Bob} (or, indeed, any neighboring world of D_I). Therefore, R should be a plausible result from any neighboring world of D_I.

Formally, D_I neighbors D_J iff $D_I = D_{J \pm x}$ for any x in the population, and:

Definition 1 A randomized query

$$Q : \mathcal{D} \rightarrow \Re^k$$

satisfies ϵ-differential privacy [4] if, for any two possible neighboring data-sets D_1, D_2 and any possible query result R:

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

Here ϵ is a small, positive value that controls the trade-off between privacy and accuracy, and is chosen by the person administering the privacy policy. To make the definition more intuitive, consider that if we set $\epsilon = ln(2)$, the above states that the result R is at most twice as likely to be produced by the true world as by any of its neighbors. Setting a smaller ϵ will provide greater privacy at the cost of additional noise, as we will demonstrate below.

The difference between the results from the true world D_1 and its neighbor D_2 is the difference the privatization noise will need to obfuscate in order for the privatized

results to not give evidence about whether D_1 or D_2 is the true world. The upper bound of this difference over $D_I \in \mathcal{D}$ is the *sensitivity* of query F.

Definition 1 The global sensitivity of a function $F : \mathcal{D} \rightarrow R^k = A$ is[1]:

$$\Delta F = \max_{D_1, D_2} \|F(D_1) - F(D_2)\|_1$$

over all pairs of neighboring data-sets D_1, D_2.

Intuitively, the sensitivity of a query is the *greatest* possible impact that adding or removing an arbitrary individual from the data-set can have on the query results, over *any* possible data-set. Suppose our analysis F asks two questions: "How many people in I are depressed?" and "How many people in I have fewer than 3 friends?" Then both answers can change by at most 1 when a single individual is added to or removed from I, and $\Delta F = 2$. If our analysis instead asks: "How many people in I are depressed?" and "How many people in I are happy?" then at most *one* answer can change by at most 1, and $\Delta F = 1$. Note that histograms, which partition the individuals of the data set into 'bucket' counts, have a sensitivity of 1: removing or adding an individual will change at most one bucket count by at most 1. This very low sensitivity makes histograms a useful tool in differentially private data-mining [4–6].

We can create a differentially private query Q by adding noise to F that is calibrated to cover up ΔF [4]:

Theorem 1 *If $F : \mathcal{D} \rightarrow \Re^k$ is a k-ary function with sensitivity ΔF then the function $F(D) + Lap^k(\Delta F/\epsilon)$ is ϵ-differentially private, where $Lap^k(\lambda)$ is a k-tuple of values sampled from a Laplacian random variable with standard deviation $\sqrt{2}\lambda$.*

The standard deviation of the Laplacian noise values is $\sqrt{2}\Delta F/\epsilon$. Thus the noise will be large if the function is very sensitive, or if ϵ is small. If we set $\epsilon = ln(2)$ on a query with sensitivity $\Delta F = 2$, the standard deviation of our added noise will be close to 4.

It is important to note that ΔF is an upper bound taken across *all possible* pairs of neighboring data-sets; it is independent of the true world. Intuitively, this is necessary because noise values which are dependent on the nature of the true world may introduce a privacy leak themselves. For example, when querying the diameter of a social network, if Alice forms the only bridge between otherwise unconnected subgraphs in the true world, removing her node and edges from the data-set causes a difference of ∞ in the graph diameter. Noise values calibrated to this true world must be arbitrarily large (and, in fact, will obliterate the utility of the result). However, consider a neighboring *possible* world including Bob, who forms a second bridge between the subgraphs (see Fig. 12); if this possible world were the true world, the difference in diameter caused by adding or removing a node would be finite, and if we calibrated the noise to that difference, it would be relatively small. If we chose

[1] The L_1-norm of $x \in \Re^n$ is defined as $\|x\|_1 = \Sigma_{i=1}^n |x_i|$.

our noise values based on the true world, an attacker could easily determine whether or not Bob was in the network: a result of $R = 300{,}453.23$ would imply Bob was absent, while the result $R = 4.23$ would indicate that Bob was present. To prevent this, global sensitivity is based on the worst-case scenario for the query, across all *possible* data-sets. In this case, this implies that diameter is a query too sensitive to be feasibly privatized using traditional differential privacy.

2.1 Smooth Sensitivity

Several sophisticated privatization techniques exist which do calibrate noise to the true data-set rather than using the worst-case upper-bound offered by global sensitivity. Consider an actual data-set D_{June12}; the *local sensitivity* of a function F on the data D_{June12} is the maximum change in F caused by removing or adding an individual from D_{June12}, analogous to computing the global sensitivity with D_1, D_2 restricted to D_{June12} and its neighboring possible worlds. In the example above, $diameter(G_{bob})$'s local sensitivity is small, while the local sensitivity of its neighbor $diameter(G_{alice})$ is very high: this jump in local sensitivities is what causes the threat to privacy described above. Since G_{alice} is created by removing one individual from G_{bob}, we will refer to G_{alice} as a *one-step neighbor* of G_{bob}, and consider a *k-step neighbor* of G_{bob} to be one created by adding or removing k individuals from G_{bob}. *Smooth sensitivity* is a technique which computes privatization noise based on both the local sensitivity of the true data-set, *and* the local sensitivity of all *k-step* neighbors scaled by k, for all k [7]. The technique 'smooths' over the local-sensitivity jumps depicted in the alice-bob graph example. However, local-sensitivity based techniques satisfy a weaker definition of differential privacy, and in some cases computing the amount of noise required to privatize a given D_I may be infeasible. We will primarily focus on techniques which satisfy strict ϵ-differential privacy in this chapter, but we will reference existing smooth-sensitivity techniques where applicable, and we recommend looking at [8] for more information on this approach.

3 Differential Privacy and Network Data

The above definition for differential privacy assumes all information about a data-set participant is provided by the participant themselves; protecting an individual's presence in the data-set then protects all the information regarding them. The situation changes when we ask survey participants to provide information about other individuals.

We will refer to individuals who contribute their knowledge to the data-set as *participants*, and individuals who have information provided *about* themselves (by others) as *subjects*. Traditional differential privacy protects participants only, and

in many cases it seems clear that subject privacy is unnecessary: if a survey counts the students who attended the "Coffee with the Dean" event, the dean's privacy is not important. By contrast, a study that counts students who report having sexual relations with the football captain exposes extremely sensitive information about its subject. Social networks are often collected from populations of interest by having participants list the full names of their friends within the population; these relationships form directed network edges leading from the participant's node to the nodes of each of their friends [9]. In this case, the friends are subjects of the participant's survey data, but the participant herself may also be the subject of some of her friends' survey data (if they also submit surveys). This presents a complex situation in which to apply differential privacy.

The core of the differential privacy guarantee is that the privatized result R is difficult to attribute to the true world versus one of its neighboring possible worlds. Adapting differential privacy to networked data amounts to deciding what we mean by "neighboring worlds" in this context. There are several possibilities; each one provides a different level of privacy guarantee and deals with a different type of "gap" between worlds. As always, there is a trade-off between privacy and utility: in general, the stronger the privacy guarantee, the more noise will be required to achieve it. We will describe two network privacy standards, *node privacy* and *edge privacy*, which have appeared in the literature.

Additionally, we propose two novel standards, *out-link privacy* and *partition-privacy*, that require less noise than existing standards; give reasonably strong guarantee of privacy similar to traditional differential privacy; and enable certain queries that required levels of noise that rendered results meaningless under existing standards.

3.1 Node Privacy

A privatized query Q satisfies *node-privacy* if it satisfies differential privacy for all pairs of graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ where $V_2 = V_1 - x$ and $E_2 = E_1 - \{(v_1, v_2) | v_1 = x \vee v_2 = x\}$ for some $x \in V_1$.

The Alice-Bob graph example in Sect. 2 implicitly assumes this privacy standard: In node privacy, if the true world is a given social network G, the neighboring possible worlds are ones in which an arbitrary node, and *all* edges connected to it, are removed from or added to G. This privacy guarantee completely protects *all* individuals, both participants and subjects. An attacker in possession of R will not be able to determine whether a person x appears in the population at all. Although this is an natural adaptation of differential privacy to social networks, it also places *extremely* severe restrictions on the queries we are able to compute, as we will demonstrate in Sect. 4, and in many cases, node-privacy may be an unnecessarily strong guarantee.

3.2 Edge Privacy

A privatized query Q satisfies *edge-privacy* if it satisfies differential privacy for all pairs of graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ where $V_1 = V_2$ and $E_2 = E_1 - E_x$ where $|E_x| = k$.

In edge privacy, if the true world is the social network G, neighboring possible worlds are ones in which k arbitrary edges are added or removed from G. An attacker in possession of R won't be able to determine with high certainty whether individuals x and y are friends, and an individual node in the graph can plausibly deny the existence of up to k of its friendships with other nodes. Single edge privacy, with $k = 1$, is the standard most often used in existing literature on differentially private graph analysis. This is a weaker guarantee than node-privacy: high-degree nodes may still have an easily identifiable effect on query results, even though their individual relationships are protected. However, this is a sufficiently strong for many applications, and enables many more types of queries to be privatized than the severely-restrictive node-privacy.

3.3 Out-Link Privacy

A privatized query Q satisfies *out-link privacy* if it satisfies differential privacy for all pairs of graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ where $V_1 = V_2$ and $E_2 = E_1 - \{(v_1, v_2)|v_1 = x\}$ for some $x \in V_1$.

This privacy guarantee protects the data contributed by data-set *participants*, using the same conceptual privacy standard as the original definition of differential privacy. Given that the true world is a social network G, the neighboring possible worlds are ones in which an arbitrary node and all of its *out-links* are removed from or added to G. An attacker in possession of R won't be able to determine whether a person x supplied their data (submitted a survey) to help produce the graph. This privacy guarantee is strictly weaker than node privacy, but compares well with single edge privacy for many queries. Any participant can plausibly deny its out-links, or, equivalently, any participant can plausibly deny one in-link from another participant node. Analogous to k-edge privacy, we can also provide k-out-link privacy by considering neighboring worlds that differ from the true world by the out-links of up to k nodes. Note that 2-out-link privacy allows two nodes to *simultaneously* deny all out-links, and as a result, this enables a complete mutual edge to be protected (providing single-edge privacy in addition to out-link privacy). In general, a k-level privacy guarantee can be satisfied by scaling the added noise by k.

Out-link privacy improves on edge-privacy by reducing the distinctive signature of high-degree nodes in the data-results, through protecting all relationships cited *by* the popular person: although others may still claim to be friends with her, she can plausibly deny those relationships are mutual. Additionally this standard simplifies sensitivity computation and noise addition, enabling many queries that would be infeasible under both node and edge privacy as we will demonstrate in Sect. 4.

3.4 Partition Privacy

Define a partitioned graph to be comprised of separate components such that $G = \{g_i\}$ for disjoint subgraphs g_i. A privatized query Q satisfies *partition privacy* if it satisfies differential privacy for all pairs of graphs G_1, G_2 where $G_1 = G_2 - g_i$ for some $g_i \in G_1$.

Many questions about social structures are naturally asked over a collection of graphs rather than one monolithic social network. Social scientists studying interpersonal interaction run experiments over large collections of small social groups, collecting social networks for each distinct group [10, 11]. Collections of disjoint social networks can be implicit in larger graphs as well. Node properties such as *dormitory, major, university*, or *geographical location* can be used to partition large graphs into meaningful sets of disjoint local social networks [12]. This enables researchers to perform tests of hypotheses about social behavior across groups, such as "Is clustering coefficient correlated with gender in dormitory friendship structures?".

This useful sub-class of analyses is especially amenable to privatization. In partition privacy, neighboring possible worlds are ones in which one subgraph is added or removed from the set of disjoint subgraphs comprising the data-set. Partition privacy is strictly stronger than node-privacy: it provides protection at the level of entire social groups rather than individuals. However, it also requires very little noise to implement. We will present a diverse selection of analyses that can be easily privatized under partition privacy.

Below we will discuss the application of these four privacy standards to common social network analysis tasks such as triangle counts (and subgraph-counts generally), degree distributions, centrality measures, graph-modeling, and other differentially privatized network analyses from the existing literature. In addition to covering previous work, we provide several infeasibility proofs and propose two original algorithms applying out-link privacy to common problems in social network analysis.

4 Applications of Differential Privacy to Social Network Analysis

We now present a straightforward guide to the application of differential privacy to several common social network analysis techniques.

4.1 Triangle Counting

Triangles, instances in which two of an individual's friends are themselves mutual friends, indicate social cohesion in the network. Triangle counts are the key parameter in the clustering coefficient, a common metric for describing and comparing graphs.

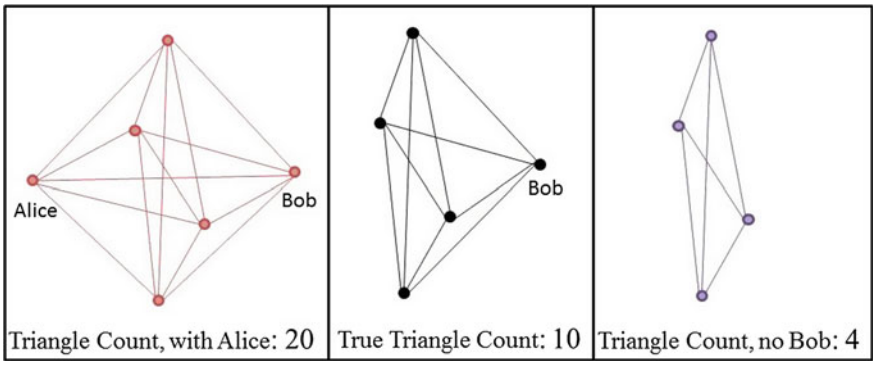


Fig. 2 Node-sensitivity of triangle-counts is a function of n, and thus is unbounded in general

Similarly, counts of other subgraphs such as stars, or squares, are used as graph statistics for graph similarity comparisons [13, 14]. All subgraph counts have similar privacy properties to the triangle count privatization described below.

Node Privacy Differentially private triangle counts are not feasible under simple node-privacy. In the worst case, adding a node to a complete graph of size n (a graph containing all possible edges), will introduce $\binom{n}{2}$ new triangles (Fig. 2). Since the change is dependent on the size of the graph, the global sensitivity of the query in general is unbounded: it is impossible to compute a finite global upper-bound (see Sect. 3).

Although the global sensitivity of the query is unbounded here, there is another approach, using ideas similar to the smooth sensitivity approach described in Sect. 2.1. If it is publicly known that the maximum degree of a graph is d, then removing or adding a node can affect the triangle count by at most $\binom{d}{2}$. And, any graph whose maximum degree is greater than d will have a k-step neighbor, for some k, whose maximum degree will be d (i.e., high-degree nodes can be removed until the maximum degree of the graph falls within the threshold). On generally sparse graphs with few nodes above degree d, the number of triangles in this bounded-degree neighbor graph will be a close approximation of the correct answer. The operation of finding the low-degree neighbor incurs its own sensitivity cost, but privacy can be still achieved at a sensitivity cost in the range $O(d^2)$ [15]. While this is untenable for large social networks, networks with low maximum degrees may successfully apply node-privacy to their triangle counts using this method.

Edge Privacy For similar reasons to node privacy, edge privacy is also not feasible for triangle-counts. In the worst case, removing an edge from a graph with n nodes can remove $n - 2$ triangles (Fig. 3). Since the sensitivity is a function of the graph size, it is unbounded in general.

However, the local sensitivity of this query under edge-privacy, the sensitivity over a specific data-set, is bounded. Consider two nodes, a and b, that have k wedges (paths of length 2) connecting them, as in Fig. 3. If G is a graph in which no pair of nodes has more than k wedges connecting them, then adding an edge to G will create

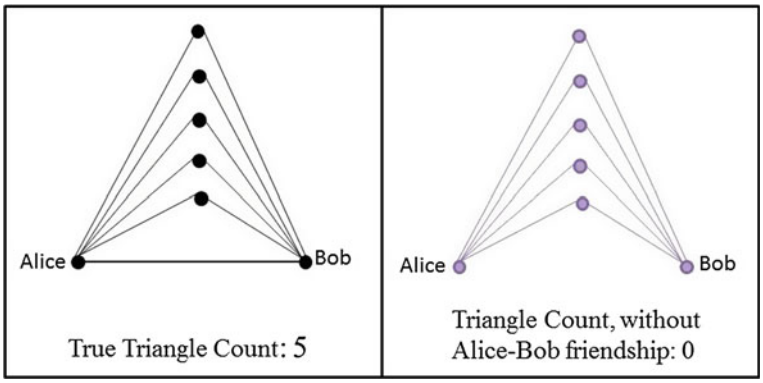


Fig. 3 Edge-sensitivity of triangle-counts is a function of n, and thus is unbounded in general

at most k triangles, and removing an edge will delete at most k triangles. We can apply smooth sensitivity techniques to take advantage of this in cases where k is not large, and thus attain a slightly weaker level of differential edge-privacy; however, real world social networks are transitive (if two people share a mutual friend, they're much more likely to be friends with each other) and this will tend to produce large values of k. When k is large, even instance-based noise addition may introduce error of a factor of 10 or greater [8].

Outlink Privacy We now propose a method for privatizing information about triangle counts and clustering coefficients under out-link privacy, using a somewhat modified version of the query that more closely mimics the information gathered from a real world social-network survey. To do this, we introduce a simple, powerful method that can be applied to gather private estimates of a variety of useful statistics over nodes in the graph.

By focusing on protecting the knowledge each individual has about their role with respect to the network, out-link privacy fits naturally with the techniques of *ego-network analysis*, an approach to social network analysis that focuses on the network as viewed by the individuals belonging to it [16]. In this approach, a network with n members is broken into n overlapping ego-network subgraphs, each consisting of a individual 'ego' node and his or her immediate neighborhood of friends (referred to as alters). A survey collecting information about the triangles in an individual's ego-network might look like Algorithm 1.

The only data that is retained by the researcher is, for each individual x: $outdegree(x)$, the number of friends the individual has, and $trianglecount(x)$, the number of triangles the individual participates in. These statistics are sufficient to determine the local clustering co-efficient of the node: the ratio between the number of triangles the node participates in and the maximum possible number of triangles for a node of that degree [13].

Out-degree and local clustering data from this survey can be collected into a two-dimensional histogram that provides detailed information about the patterns of social

Algorithm 1 A survey gathering information about triangles

function TRIANGLEQUERY
  $friendlist \leftarrow$ Query("Who are your friends?")
  $friendpairs \leftarrow$ CrossProduct($friendlist$, $friendlist$)
  $outdegree \leftarrow$ Size($friendlist$)

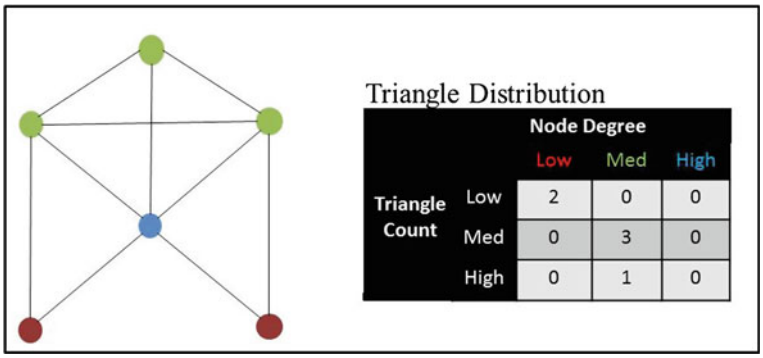
  $triangles \leftarrow$ Query("Which of these pairs are friends with each other?", $friendpairs$)
  $trianglecount \leftarrow$ Size($triangles$)
  return ($outdegree$, $trianglecount$)
end function

cohesion of the graph and has a very low sensitivity under out-link privacy: removing or adding an individual's survey data to the histogram only alters one partition count by at most one, and thus the noise required to privatize this data-structure would be very small. Histograms with fewer partitions and larger count values in each partition are less sensitive to added noise; we propose Algorithm 2 that produces a very flexible, robust, and safely privatized representation of the social cohesion patterns in the network using local triangle counts.

Algorithm 2 Privatizing local clustering coefficient distribution data

function PRIVATECLUSTERING(deg_{low}, deg_{med}, $data$)
  Initialize($bins$[][])
  for all ($nodeDegree$, $triangleCount$) $\in$ $data$ **do**
    $degBin \leftarrow$Partition($nodeDegree$, deg_{low}, deg_{med})
    $localCluster \leftarrow triangleCount/(nodeDegree * (nodeDegree - 1))$
    $triBin \leftarrow$Partition($localCluster$, 1/3, 2/3)
    $bin[degBin][triBin] \leftarrow bin[degBin][triBin] + 1$
  end for
  for $i = 0 \rightarrow 2$, $j = 0 \rightarrow 2$ **do**
    $bins[i][j] \leftarrow bins[i][j]+$ LaplacianNoise(1)
  end for
  return $bins$
end function

Algorithm 2 takes as input two node-degree threshold values, deg_{low}, deg_{med} and uses these to partition the ($outdegree$, $trianglecount$) data-points collected from the $TriangleQuery$ survey into low, medium and high degree nodes. The algorithm then computes the local clustering coefficient of each node and further partitions nodes by these values, creating a histogram with nine partitions (see Fig. 4). Laplacian noise sufficient to cover a function sensitivity of 1 is added to each partition, and the privatized result may be released. We can consider the effect of this noise in terms of how many of the noisy, privatized partition counts can be expected to differ measurably from their true values. With only nine counts and a sensitivity of 1, the expected number of privatized partition counts that will differ from their true values by more than 3, is less than 0.25. The released histogram accurately and succinctly



Triangle Distribution

Triangle Count	Node Degree: Low	Node Degree: Med	Node Degree: High
Low	2	0	0
Med	0	3	0
High	0	1	0

Fig. 4 The triangle distribution allows us to present clustering information with an out-link sensitivity of 1

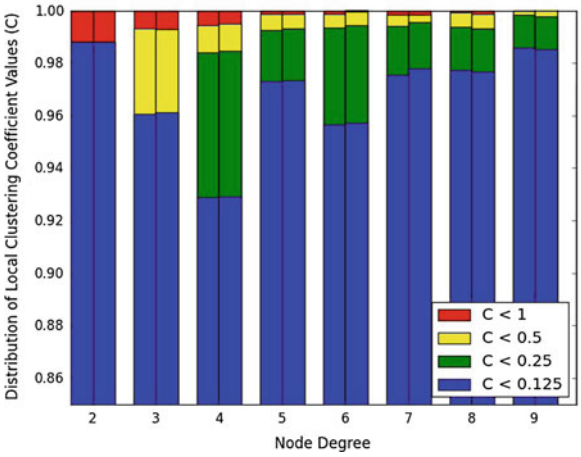


Fig. 5 A comparison of true and privatized results over the Slashdot Zoo social network

captures useful information about the distribution of meaningful local patterns across the graph.

Figure 5 shows the effect of privatization noise on the outlink-private triangle counting algorithm over the Slashdot 'Zoo' social network [17]. Here, nodes are binned by their degree and their local clustering coefficient, and the resulting degree-bins are normalized to show the distribution of clustering coefficients in each degree bin. The true counts are given in the left columns, their privatized versions are given in the right. Note that, as the magnitude of noise added by this algorithm is very small in comparison to the scale of the data, it was necessary to focus on a small section of the results in order for the effect of the noise to be visible.

The same simple approach can be used to collect and privatize any information available within an ego-network, simply by restructuring the survey appropriately. For example, replacing question 2 in the survey of 1 by the question "For each of

your friends, add a check mark if the two of you share at least one additional, mutual friend" will collect information about the probability that an edge participates in a triangle. The question "Are you part of a group of at least k friends who are all mutual friends with each other?" collects statistics about cliques in the graph.

If undirected social network data must be privatized, the survey-collection approach described above may be simulated by considering each node's immediate neighborhood as their ego-network view, and sub-sampling by introducing α probability that the ego is unaware of any given edge between its alters.

Partition Privacy In applications that require a collection of disjoint social networks, even more detailed privatized analysis is possible. Partition-privacy allows essentially arbitrary analysis of individual graphs in the data-set and then privatizes the aggregation of the independent results. Assume an analysis has been performed on each individual graph, producing either a numerical result with a publicly known range (e.g., the global clustering coefficient of the graph), a category result (the gender of the dorm represented by the graph), or any combination of numerical and categorical results. The collection of graphs may now be viewed as a collection of multi-attribute data points. Removing or adding one graph from the collection is equivalent to removing or adding one of these data points; we can apply traditional differential privacy techniques to this set of independent data points as though we were working with tabular data over individuals. Two low-sensitivity techniques are very useful here: histograms and privatized means. We will demonstrate the application of these techniques in the examples below, beginning with an application of partition privacy to triangle-count data.

The global clustering coefficient is the proportion of wedges in the graph (where one person has a pair of friends) that are closed to form a triangle (i.e., the pair of friends are also friends with each other); formally, $Clustering\ Coefficient(G) = \frac{3*[number\ of\ triangles\ in\ G]}{[number\ of\ wedges\ in\ G]}$. A graph with no triangles has a clustering coefficient of 0; a clique has a clustering coefficient of 1. The clustering coefficient of a graph is a useful normalized measure of its social cohesion. However, it is difficult to draw meaningful conclusions about the population being studied using one piece of data in isolation. Given a collection of social networks, we can identify meaningful patterns of behavior by comparing clustering coefficients across networks.

Assume we want to examine how attribute X of a social group affects its degree of social cohesion. For example, we could study the relationship between the gender of a college dormitory and the clustering coefficient of the social network within the dorm (see Fig. 6). Given a data-set consisting of a collection of social networks for each possible value of X (e.g., a set of male, female and co-ed dorms), we first compute the global clustering coefficient over each individual network. We can then compute the mean of the clustering coefficients for each value of the attribute X, add noise to privatize the result, and release the privatized means.

The mean of a set of bounded numerical values has low sensitivity when the number of values is publicly known. Consider the mean $MaleDormsClustering = M/N$ where $M = \Sigma_{G \in MaleDorms}\, clustering\_coefficient(G)$ and N is the number of male-only dorms in the data-set. If N is publicly known (for instance, because

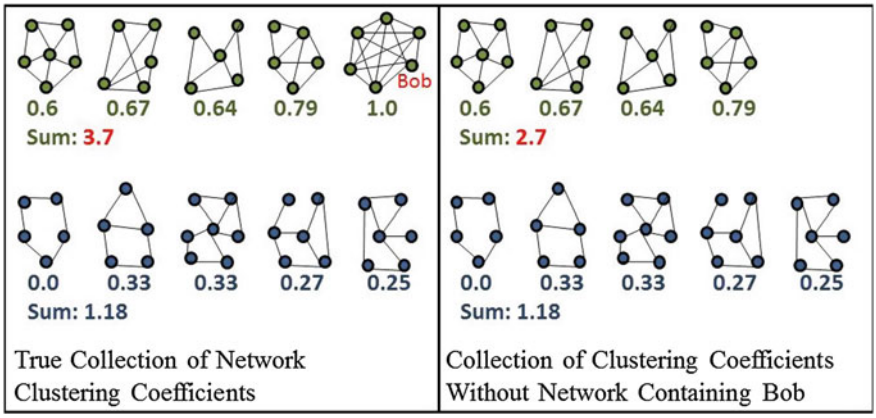


Fig. 6 Removing or altering one graph from the partitioned graph set only affects the numerator of the mean by one

each university's dorms are listed on their website) we can safely skip adding noise to this value and focus on privatizing only the numerator M without reducing the privacy of the result [18]. Since M is a sum of clustering coefficients that have values in the bounded range [0, 1], adding, removing or altering one clustering coefficient will alter the sum M by at most 1. Thus the sensitivity of the sum M is 1, and the value $\frac{M+Lap(1/\epsilon)}{N}$ will be differentially private. Note that the noise added to the true values of $MaleDormsClustering$ has a standard deviation of only $Lap(1/\epsilon)/N$.

4.2 Degree Distribution

The degree distribution of a graph is a histogram partitioning the nodes in the graph by their degree; it is often used to describe the underlying structure of social networks for purposes of developing graph models and making similarity comparisons between graphs [19].

Node Privacy Although degree distributions are represented as histograms, the sensitivity is not small under node privacy because one node affects multiple counts in the distribution: removing a node from the graph reduces the degree of all nodes connected to it. A node with k edges can affect a total of $2k + 1$ values of the distribution (Fig. 7). In the worst case, adding a node of maximal degree will change $2n + 1$ values, and since this sensitivity is dependent on n, it will be unbounded in general (see Sect. 3).

Edge Privacy Edge privacy is feasible for degree distributions. Removing one edge from the graph changes the degree of two nodes, and affects at most four counts (Fig. 8). Under k-edge privacy, the sensitivity is $4k$. With a sufficiently large graph,

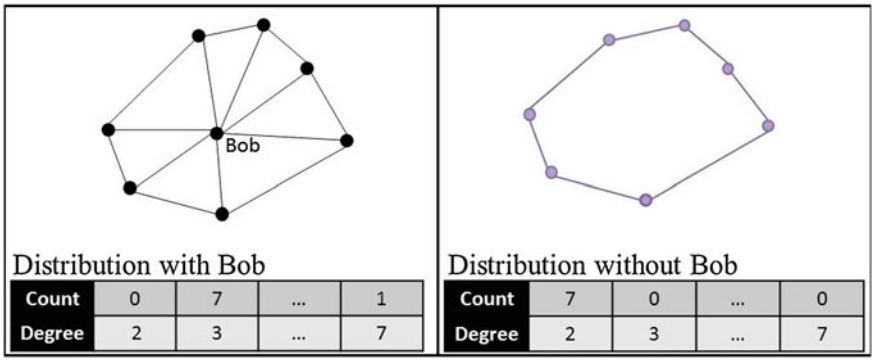


Fig. 7 Node sensitivity of degree distribution queries is a function of n, and thus is unbounded in general

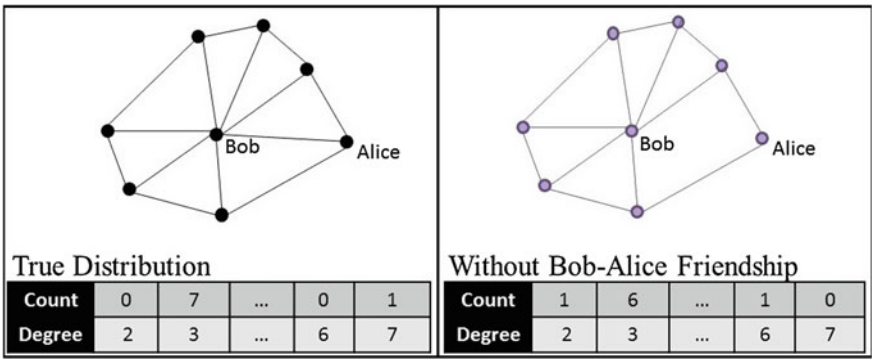


Fig. 8 Edge sensitivity of degree distribution queries is 4: at most four values can change by one when a node is added or removed

this is a negligible amount of noise, and the utility of this technique has been successfully demonstrated [6].

Outlink Privacy Out-link privacy, in contexts where it is deemed sufficient, requires even less noise for degree distributions. Here, we consider just the distribution of out-degrees, the result of asking participants, "How many friends do you have?" Removing one node and its out-links from the graph affects only one value in the degree distribution (Fig. 9). Under this privacy standard, a high-degree node may still leave evidence of its presence in the data-set through the out-degrees of its friends. However, there are many possible explanations for a slightly higher-than-expected degree among nodes in the graph: they may represent additional friendships among the nodes, or outside friendships with individuals who were non-participants in the survey. Exploiting this vulnerability to guess the presence of a high-degree node with any certainty would require an attacker to possess near complete information about the true social network.

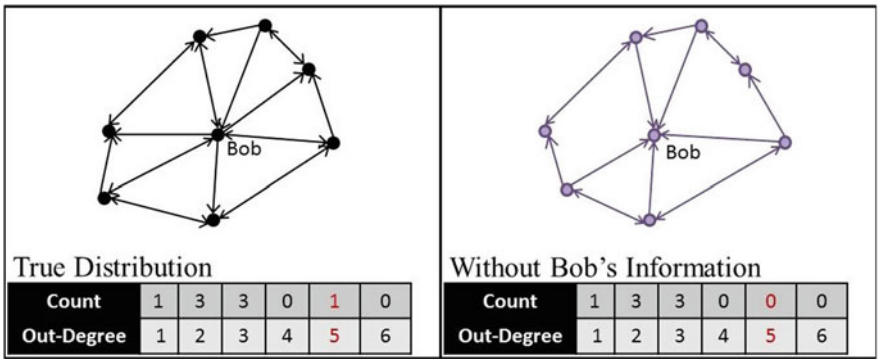


Fig. 9 Out-link sensitivity = 1. Protecting the out-edges of a node provides privacy with relatively little effect on the degree distribution

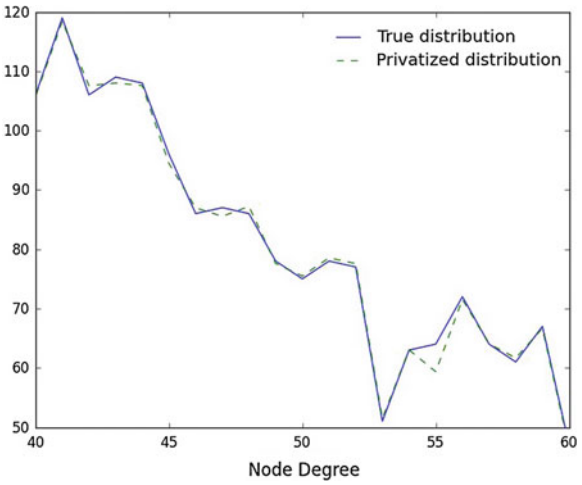


Fig. 10 A comparison of true and privatized results over the Slashdot Zoo social network

Figure 10 shows the effect of privatization noise on the outlink-private degree distribution algorithm over the Slashdot 'Zoo' social network [17]. Again, as the noise added by this algorithm is very small, the figure focuses on a subsection of the results in order to make the effect of the noise visible.

Partition Privacy Partition privacy can also enable privatized analysis of degree distribution data. Consider the context in which a researcher performs an experiment to directly study behavior patterns in small social groups. A common technique is to assign people to small groups where they must work cooperatively to solve problems [10, 11]. Interpersonal communications in each group are monitored and analyzed. Raw communication data can be transformed into social network graphs by adding edges between nodes that communicate frequently. In small groups, different degree

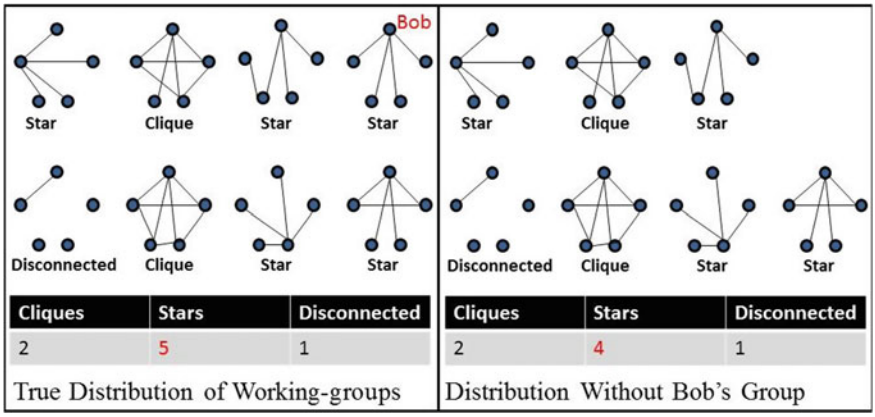


Fig. 11 Removing or adding one graph only affects the count in one histogram category by one

distributions will indicate different patterns of cooperation; for example, groups may have one high-degree 'leader' centralizing communication, or they might cooperate equitably together producing a near clique graph (see Fig. 11). These degree-distribution categories may be affected by the group's context (e.g., working in person, or online), and they may affect the group's performance on the assigned task. When degree-distributions help us attach a meaningful category label to individual networks, we can use a privatized histogram to safely release the distribution of these labels across the set of networks. If desired, we can further partition this histogram using properties such as the group's context or performance score to create more informative multi-dimensional histograms (for an example of a multi-dimensional histogram, see Fig. 14). As described in Sect. 2, histograms have a sensitivity of only 1 and may be safely released by adding Laplacian noise calibrated to that sensitivity to each count.

4.3 Centrality and Paths

Centrality measures attempt to gauge the relative "importance" of specific individuals within the social network; they may be studied on a per-node basis, identifying influential members of the community, or as distribution scores providing information about the overall behavior of the social network [20]. The simplest centrality measure is node degree: nodes with high degree are more likely to be influential in the network. However, other centrality measures take into account information from across the network: *betweenness* scores individuals by the number of shortest-paths between other pairs of nodes across the network that pass through them, and *closeness* scores nodes by the sum of their distances to all other nodes in the graph.

The two more complex centrality measures present difficulties for traditional approaches to differential privacy in social networks. Clearly, it is impossible to release a named list of influential individuals under node-privacy. But even distrib-

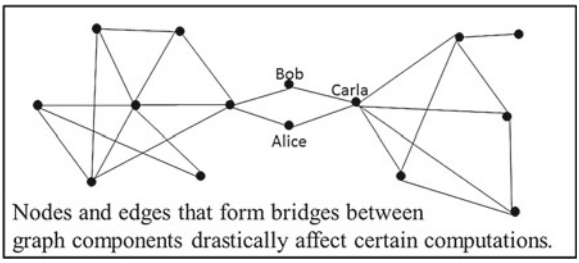


Fig. 12 Removing one node or edge from a graph can change path lengths catastrophically

utions of centrality scores can be very sensitive, under both node and edge privacy, due to the role of bridges in the graph. Removing a node, or edge, that forms the only connection between two otherwise disconnected subgraphs will have a catastrophic affect on path distances in the network, causing finite distances to become infinite, and thus will drastically alter betweenness and closeness scores (see Fig. 12). In general, privatizing traditional centrality measures, or any metric that relies on path lengths, remains an open problem under differential privacy.

Outlink Privacy We propose a very different approach for collecting and privatizing information about influential nodes within a network; one that satisfies out-link privacy (by protecting individuals' data contributions) and leverages individuals' knowledge about their community. We define a *popularity graph*: a synthetic network that represents the social structure among influential community members (Algorithm 3).

Individuals in the population are asked to "list up to three of your most popular friends within the specified population group". A base graph is created containing nodes for all members of the population group, and undirected edges of weight 0 are added between all pairs of nodes. The data collected from the survey is then added to the graph: when two popular people are listed on the same survey, the weight of the edge connecting them is incremented. For example, if a person submits a survey listing three popular friends, weights of every edge in the triangle connecting those friends will be incremented. The sensitivity of the popularity graph is 3, since a maximum of 3 edge-weight values can change if a participant adds or retracts their data.

To privatize the data, appropriate Laplacian noise to cover a function sensitivity of 3 is added to all edge-weights. Then two post-processing steps are applied: edges with low weight are eliminated, and the graph is anonymized. The resulting weighted popularity graph is published (Fig. 13). This graph can be used to understand the underlying social influence structure of the population, identifying social clusters and the bridges between them. The privacy of data provided by the query participants is fully protected; however, the subjects who appear as nodes in the graph will clearly be less secure and this analysis may not be appropriate in all contexts. For many population though, the popularity graph should be sufficient protection: anonymity, noisy edges, and the fact that the artificially-constructed graph will lack detailed substructures often used for

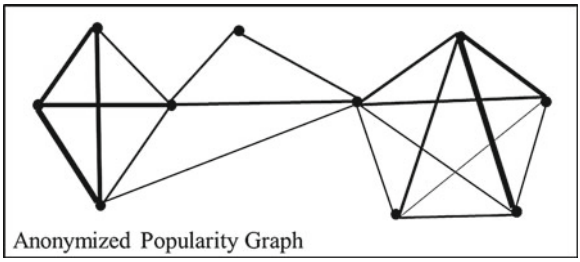


Fig. 13 A popularity graph with edge thickness indicating edge-weight

re-identification attacks, will all contribute to protecting the privacy of the query subjects.

Algorithm 3 Privatizing centrality data

```
function PRIVATECENTRALITY(importanceT, data_I)
    V ← population
    E[i][j] ← 0 ∀i, j ∈ V
    for all i ∈ I do
        ∀p_j, p_k∈data_I[i], E[p_j, p_k]←E[p_j, p_k] + 1
    end for
    for all i, j ∈ population do
        E[i, j] ← E[i, j]+ LaplacianNoise(3)
        if E[i, j] < importanceT then
            E[i, j] ← 0
        end if
    end for
    return PopularityGraph = (V, E)
end function
```

Partition Privacy A noteworthy property of partition privacy is that it does not exhibit the high sensitivity to path length queries that constrains other forms of graph privacy. Although removing a bridge will drastically affect path lengths in a given network, it will only affect *one* network in the collection of small disjoint networks that comprises the data-set for a partition privacy application. This enables privatized analysis for a wide variety of graph properties that are otherwise too revealing to be released.

The average shortest-path distance for a network is a measure of its connectedness. Given a collection of networks, we can find the average shortest-path length for each network and aggregate the results into a histogram, giving us information about the patterns of graph-connectedness across our data-set (see Fig. 14). As the sensitivity of a histogram is just 1, the results can be privatized by adding a relatively small amount of noise to each count. The same technique can be used on any numerical or categorical graph property: we can privatize the distribution of maximum centrality

Fig. 14 With a set of graphs, histograms can be used to release information about the relationships between multiple variables, including path lengths, with low sensitivity

scores, number of bridges per graph, or even graph diameters. This flexibility of application is one of the primary advantages of partition privacy.

4.4 Graph-Modeling and Social Recommendations

Several groups have proposed differentially private approaches to creating graph models–randomized synthetic graphs that are generated to be similar to a true, private, social network and thus can be studied safely in place of the original graph. The Stochastic Kronecker graph model has been privatized under edge-privacy [21], and several other groups have developed their own models that satisfy differential edge privacy [22–24].

We also note that the results from our proposed out-link privatized degree distribution and triangle statistics (see Sects 4.1, 4.2) could provide privatized input for the Transitive Chung Lu graph model proposed by Pfeiffer et al. [25]. This model is somewhat unique in the literature for its ability to generate graphs that match both the degree distribution and clustering coefficient of the original target graph.

Finally, the possibilities and difficulties of applying edge-privacy standards to social network recommendation systems are explored in [26].

5 Conclusions

Differential privacy represents a potentially powerful tool for analyzing social networks while providing strong guarantees of privacy for individual participants. The application of differential-privacy guarantees to social-network analysis allows results to be released with confidence that individual data will not be compromised by malicious attackers, even with the benefit of arbitrary background knowledge.

By providing this guide to differentially private social network analysis, along with new, powerful techniques for privatizing social-network data, we hope to spur the application of these standards to social-network data in a practical fashion. In future work we plan to study the application of out-link privacy and partition privacy to other social-network analysis tasks and provide studies of these approaches on real-world network data.

Acknowledgments This work was supported by the Center for the Science of Information, an NSF Science and Technology Center.

References

1. Narayanan A, Shmatikov V (2008) Robust de-anonymization of large sparse datasets. In: Proceedings of the 2008 IEEE symposium on security and privacy, pp 111–125
2. Zheleva E, Getoor L (2011) Privacy in social networks: a survey. In: Aggarwal CC (ed) Social network data analytics, p 277
3. Narayanan A, Shmatikov V (2009) De-anonymizing social networks. In: 2009 30th IEEE symposium on security and privacy, pp 173–187
4. Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. In: Proceedings of the 3rd theory of cryptographyconference. pp 265–284
5. Hay M, Rastogi V, Miklau G, Suciu D (2010) Boosting the accuracy of differentially private histograms through consistency. Proc VLDB Endow 3(1–2):1021–1032
6. Hay M, Li C, Miklau G, Jensen D (2009) Accurate estimation of the degree distribution of private networks. In: IEEE international conference on data mining, pp 169–178
7. Nissim K, Raskhodnikova S, Smith A (2007) Smooth sensitivity and sampling in private data analysis. In: Proceedings of the thirty-ninth annualACM symposium on Theory of computing. ACM
8. Karwa V, Raskhodnikova S, Smith A, Yaroslavtsev G (2011) Private analysis of graph structure. In: Proceedings of the VLDB Endowment, vol 4(11)
9. Marsden P (1990) Network data and measurement. Annu Rev Sociol 435–463
10. Sparrowe RT, Liden RC, Wayne SJ et al (2001) Social networks and the performance of individuals and groups. Acad Manage J 44:316–325
11. Gladstein DL, Reilly NP (1985) Group decision-making under threat-the tycoon game. Acad Manage J 28:613–627
12. Traud AL, Mucha PJ, Porter MA (2011) Social structure of facebook networks. Physica A 391:4165–4180
13. Watts DJ, Strogatz SH (1998) Collective dynamics of "small-world" networks. Nature 393(6684):440–442
14. Holland P, Leinhardt S (1976) Local structure in social networks. Sociol Method 7(1)
15. Blocki J, Blum A, Datta A, Sheffet O (2012) Differentially private data analysis of social networks via restricted sensitivity. CoRR abs/1208.4586

16. Marin A, Wellman B (2010) Social network analysis: an introduction. In: Handbook of social network analysis, p 22
17. Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2008) Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. CoRR abs/0810.1355
18. Christine Task CC, Publicly constrained populations in differential privacy
19. Newman M (2003) The structure and function of complex networks. SIAM Rev 167–256
20. Degenne A, Forsé M (1999) Introducing social networks. SAGE Publications Ltd, New York
21. Mir DJ, Wright RN (2009) A differentially private graph estimator. In: Proceedings of the 2009 IEEE international conference on data mining workshops. IEEE Computer Society, pp 122–129
22. Proserpio D, Goldberg S, McSherry F (2012) A workflow for differentially-private graph synthesis
23. Sala A, Zhao X, Wilso C, Zheng H, Zhao BY (2011) Sharing graphs using differentially private graph models. In: Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference, New York, NY, USA, ACM, pp 81–98
24. Gupta A, Roth A, Ullman J (2012) Iterative constructions and private data release. In: TCC, pp 339–356
25. Pfeiffer III PP, Fond TL, Moreno S, Neville J (2012) Fast generation of large scale social networks with clustering. CoRR
26. Machanavajjhala A, Korolova A, Sarma AD (2011) Personalized social recommendations: accurate or private. Proc VLDB Endow 4(7):440–450