Private Link Prediction in Social Networks

Rediet Abebe and Vasileios Nakos

December 8, 2014

Abstract

In this work, we investigate the link prediction problem on social networks presented by Liben-Nowell and Kleinberg [8] in the context of differential privacy. In particular, after running three link prediction algorithms–common neighbors, Jaccard's coefficient, and preferential attachment–we investigate whether it is possible to answer graph cut queries in a differentially private manner. The graph cuts that we consider are arbitrary but fixed. We give positive results for privacy guarantee using the notion of restricted sensitivity under the hypothesis that our networks have bounded degree. We also present some experimental results to show the performance and utility of our methods.

Contents

1 Introduction

Social networks are often modeled as graphs, where the nodes correspond to people and edges to social interactions. Such networks are dynamic; they evolve over time to add new nodes and edges. Understanding the dynamics which drives the evolution of these networks is a complex problem, but we can ask a simpler question that can give us insight into their underlying structure. Given two nodes, we can focus on predicting the likelihood of an edge existing between them in the future. This is known as the *link prediction problem.*

The link prediction problem was popularized by Liben-Nowell and Kleinberg in [8]. In this paper, they specifically focus on co-authorship graphs extracted from arXiv, although their methods can be transferred to other social networks. They consider five particular subsections of arXiv and look at a snapshot of these graphs during the time-interval $[1994, 1996]$. i.e., they only see new edges that occur in this time-frame, and try to infer which new edges are likely to occur in the time-interval $[1997, 1999]$. The former is referred to as the training-interval, and we denote the corresponding graph by G_1; the latter is the test-interval, with corresponding graph G_2. The construction of these graphs makes it such that G_1 and G_2 have the same node-set but disjoint edge-sets. The estimate of G_2 obtained by using various link prediction techniques is denoted by G_p. The link prediction problem is important in this domain since a collaboration between two influential researchers in different fields could pioneer exciting new directions.

In [8], Liben-Nowell and Kleinberg perform large-scale experiments to show that link prediction is dependent on the topology of the network by using subtle measures for detecting node proximity of nodes. They use various techniques adapted from graph theory and network analysis and present strong results showing that several methods outperform random predictions. In particular, measures based on clustering and low-rank approximation consistently perform well. However, their results also show that there is no overall clear winner for the best link prediction method, hinting at the fact that which link prediction algorithm is best to use might be dependent on the type of graph we are looking at as well as the baseline for performance.

There are various techniques for link prediction ranging from feature-based classification and kernel-based methods to matrix factorization and probabilistic graphical models, which are surveyed in [1]. These methods differ in a lot of ways, including complexity, prediction performance, scalability, and generalization ability. Some might be more inherently suitable for doing link prediction for classes of networks exhibiting certain properties than others. The methods considered in [8] include, graph distance, common neighbors, Jaccard's coefficient, Adamic/Adar, preferential attachment, hitting time, commute time, PageRank, etc. In this paper, we are specifically interested in common neighbors, Jaccard's coefficient, and preferential attachment, which we will discuss in more detail in Section 3.

While problems such as link prediction, community detection, homophily, etc, have been studied extensively, there is a hole in the literature on social networks–there is comparatively little work related to privacy. Since co-author graphs are publicly available,

links in these networks do not encode sensitive information. However, there are various other social networks for which privacy is crucial. The canonical example is the sex graph, where the nodes signify people and edges signify sexual relations between the corresponding nodes. Assume it is known that node u is diagnosed with an STD during time t, and we want to predict how many new links appear between u and other nodes in a later time-frame. We want to answer such a query without compromising the privacy of other edges. Clearly, sex graphs are not as widely available as co-authorship graphs. However, we can consider equivalent questions in more publicly available graphs and translate privacy guarantees back to the former case.

In this project, we seek to find privacy results related to the link prediction problem. For experimental results, we work with the same co-authorship graphs as in [8]. The notion of privacy that we consider here is edge-specific. That is, the node-set is public knowledge whereas we seek to keep the existence or non-existence of edges between any two nodes private. In Section 4, we present some attempts as well as positive results. We restrict ourselves to a cut query for a specific partition of the given graph (such as collaborations between American and non-Americans, men and women, and so on). We show that it is possible to implement the common neighbors, Jaccard's coefficient, and preferential attachment algorithms to answer such questions while guaranteeing ϵ-differential privacy. To do so, we use the notion of restricted sensitivity from [4], which allows us to use prior knowledge about the graph (that it is bounded degree) to bypass issues that we would have encountered if we had used local or global sensitivity. In light of these privacy guarantees, we also explore whether we get reasonable utility guarantee in some of these cases.

2 Related Work

There are several introductions to the field of social networks which overview various topics, two notable ones being [6] and [10]. While they give a comprehensive discussion of problems including cascades, influence-maximization, network formations, and games on networks, they do not address privacy extensively. With the accessibility of large scale datasets and the increase in focus on social networks in recent years, there is an emerging subfield of privacy in social networks.

One example is [12], a report on "Big Data and Privacy" presented to the President's Council of Advisors on Science and Technology (PCAST). In this report, they address privacy issues in various big-data contexts including social network analysis. This field has, in recent years, greatly benefited from the growth and increasing accessibility of large-scale digital data through, for example, open application programming interfaces to online social-media platforms. It has also been yielding insightful results about how people interact, how cascades spread, what friendship circles can tell us about individuals, and so forth.

Link analysis accounts for a size-able portion of the results in social networks. For instance, the existence of links can tell us about the adjacent nodes, at times even more

so than if we used standard measures for gathering this information. Take [3] and [9], for instance. These papers focus on how much one can infer about individual nodes based on the attributes of their neighboring nodes. In [3], they analyze the relationship between geographic location of individuals and their friends' locations. Using this, they are able to create an algorithm to predict the geographic location of an individual user based on the location of a small number of friends in their networks with higher accuracy than they if they were simply looking at the user's IP address. In general, these papers show that some attributes can be inferred with high accuracy when given information about very few nodes on the network.

The local network structure of each node is also shown to form a "unique signature" of that node in [2]. That is, the structure of an individual's network is likely unique and it is possible to identify individuals more easily from looking at their network structure than from a database analysis alone, even in anonymized social networks. This is shown to be used to formulate a family of attacks such that it is possible for an adversary to learn whether edges exist or not between specific targeted pairs of nodes. Inferring links in such networks also continues to be an active area of research.

Link analysis, and in particular link prediction, has wide-reaching and stunning applications. Another canonical example is terrorist or criminal networks, whose dynamics might be different from more public social networks. Understanding the structure of such networks and being able to infer missing links can be useful in preventing crimes, capturing criminals, and protecting national security. In less sensitive contexts, there is also recent investigation on the extent to which social ties can be inferred from co-occurence in time and space, as reveled through pictures, videos, and RSVPs [13]. A comprehensive survey of link prediction techniques can be found in [1]. In this paper, we focus our attention to fundamental ones discussed in [8] in the context of co-authorship graphs.

The privacy framework that we work under is differential privacy. While there has been significant progress in multiple domains including database query-release problems, mechanism design, and machine learning, there is not as much progress on releasing useful statistics about graph or network data while providing rigorous privacy guarantees. One of the most related piece of work is [7], which is inspired by [11], where they look at subgraph counting queries while preserving edge privacy. That is, while the nods of the graph are public, they preserve the privacy of the presence or absence of edges. They give algorithms that satisfy a strong notion of privacy and give good estimates to these subgraph counting queries.

In this paper, we use a tool (called restricted sensitivity) from [4], which improves accuracy in differentially private data analysis by taking advantage of any belief about the dataset the analyst might have. The belief that we consider in this paper, in particular, is that these networks are of bounded degree and the analyst has a rough estimate for the maximum degree of this network.

3 Setup and Data

The social networks we consider are represented using unweighted, undirected graphs $G = \langle V, E \rangle$, where V denotes the set of individuals and E relations between them. In this paper, we assume that the nodes are public knowledge and that all networks are over the same node-set. Given a vertex $v \in V$, let $\Gamma(v)$ denote the neighbors of v, i.e., the set of nodes adjacent to it.

We consider co-authorship graphs from five sections of arXiv–astrophysics, condensed matter physics, general relativity and quantum cosmology, high energy physics-phenomenology, and high energy physics-theory–between two time intervals $[1994, 1996]$ and $[1997, 1999]$. We denote the graph from the first time interval by G_1, that from the second interval by G_2, and the prediction of G_2 by G_p. As in [8], the nodes that we consider are those corresponding to authors that have at least 3 collaborators in both time intervals. This is to minimize noise since nodes with fewer neighbors are likely not crucial to the network.[1] The set of nodes satisfying this restriction form the core set.

We present the sizes of the graphs reported in [8] and those that we extracted and will use in our experiments both with and without the core restriction below for comparison:

	training period			Core		
	authors	papers	edges	authors	$\|E_{old}\|$	$\|E_{new}\|$
astro-ph	5343	5816	41852	1561	6178	5751
cond-mat	5469	6700	19881	1253	1899	1150
gr-qc	2122	3287	5724	486	519	400
hep-ph	5414	10254	17806	1790	6654	3294
hep-th	5241	9498	15842	1438	2311	1576

Figure 1: Size of arXiv Graphs Reported in [8]

Section	**Training Node**	**Training Edge**	**Core Node**	**Core Edge**
astro-ph	7000	22419	2651	17754
cond-mat	7858	12810	1297	3506
gr-gc	2916	3284	339	734
hep-ph	756	2128	464	938
hep-th	7091	12737	5050	5083

[1] In this particular paper, we are also limited in computational power and making such a restriction is necessary to be able to execute some of our computations.

Note that in several of these cases, the sizes of the graphs that we extracted are close to the sizes of the graphs reported in [8]. There is some noise associated with the extraction process, which is reported in [8] as well. For example, the authors are identified using their last name and first name initial, which creates duplicity issues. Some of the articles are also updated at a later point to include middle name initials, and might turn up as different authors in our case. In addition, there may have been removal of some articles from that time-frame, although we suspect that this does not account for much of the difference in sizes. The difference in size between the graphs that we work with and theirs can perhaps also be a result of the fact that we process strings differently. In some of these cases, such as high energy physics-phenomenology, we do not consider the entire graph due to computation-limitations so we only consider a subinstance.

3.1 Link Prediction Methods

In this section, we describe the metrics that we will use for measuring edge likelihood:

1. **Common neighbors**: the metric here is,

$$\text{score}(x, y) = |\Gamma(x) \cap \Gamma(y)|.$$

   The idea this metric tries to capture is that two nodes that are adjacent (i.e., co-authors, friends, etc), will introduce their neighbors to one another so future interaction with nodes adjacent to one's neighbors' is more likely than interaction with random nodes. It is shown in [10] that this value is highly correlated with future collaboration in the context of collaboration networks.

2. **Jaccard's coefficient**: the metric is,

$$\text{score}(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}.$$

   This can be viewed as a normalization of the common neighbors. This coefficient is a similarity metric used in information retrieval to measure the probability that x and y share a feature that at least one of them has, which is co-authorship in this case.

3. **Preferential attachment**: this is given by,

$$\text{score}(x, y) = |\Gamma(x)| \cdot |\Gamma(y)|.$$

   This captures the idea that new edges involving a vertex form with probability proportional to the size of the neighborhood of that vertex. This forms highly clustered graphs with a few nodes of very large degree. (By symmetry, nodes with high degree are more likely to form relations with other nodes of high degree.) As a side

note, this metric was actually initially proposed based on empirical analysis of the co-authorship graph.

We follow the method used in [8]: we run the link prediction algorithm on the training interval G_1. Each method will assign a weight score(x, y) to each pair of nodes (x, y). We list these in decreasing order, where the node pairs near the top are more likely to appear while those near the bottom are less likely to appear according to the link prediction algorithm. Assume that the true graph during the test interval has n edges. Then, the predicted graph is that formed by the top n node-pairs in this list. Note that, for our purposes, we can use any threshold to truncate this list and thus will use the edges corresponding to the top 1000 scores. This is essential in being able to execute some of our computations on larger graphs.

3.2 Differential Privacy

As mentioned above, we work in the differential privacy setting. Let $\mathcal{D}$ be the set of all possible datasets, which in this case is networks. We can restrict ourselves to a subset of this dataset $\mathcal{D}' \subseteq \mathcal{D}$, which share some property. The curator has full knowledge and access to these networks and would like analysts to make use of it. However, we would also like to release useful statistics on these networks without compromising the privacy of any of the individuals in the sense that we will define below.

In differential privacy, we are concerned with datasets that are almost the same, i.e., neighboring datasets, for the purposes of ensuring protection of any single item. Defining neighboring datasets is more subtle in the case of networks than in other forms of datasets as the item is not as unambiguously defined. There are two popular notions: edge-adjacency and vertex-adjacency.

Definition 1 (Edge-adjacency)**.** We say that two networks G and G' on the same node-set are neighboring, denoted by $G \sim G'$, if the symmetric difference of their edge set contains exactly one edge.

Vertex-adjacency, on the other hand, implies that there exists a single vertex whose removal from both networks would result in the remainder of the networks being the same. In the general setting, vertex-adjacency implies edge-adjacency. However, differential privacy results using vertex-adjacency are difficult to obtain and since it is conceivable to assume that the node-set of the networks we are considering is public knowledge, we will only consider edge-adjacency.

In this paper, we pursue two different directions: in the first one, presented in Section 4.1, we consider our database to consist of single graphs and consider two datasets to be neighboring if and only if they differ on exactly one edge after we employ the core restriction on them. In the second case, which we present in Section 4.2, the dataset consists of pairs of graphs (G_1, G_2) from two different time-frames. Recall that these graphs are on the

same node-set but have disjoint edges-sets. We say that two such pairs are adjacent, i.e., $(G_1, G_2) \sim (G_1', G_2')$, if and only if $G_1 \sim G_1'$ and $G_2 \sim G_2'$.

Most of the work on differential privacy concerns itself with datasets that are pairwise neighboring. For the tools that we use below, we can consider longer chains of neighboring datasets. In particular, we define the distance $d(D, D')$ between these two datasets to be the minimal non-negative integer ℓ such that there exits a path,

$$D = D_0 \sim D_1 \sim, \cdots, \sim D_\ell = D'.$$

Additionally, given $\mathcal{D}' \subset \mathcal{D}$, the distance of a database D to $\mathcal{D}'$ is,

$$d(D, \mathcal{D}') = \min_{D' \in \mathcal{D}'} d(D, D').$$

Definition 2. A mechanism A is (ϵ, δ)**-differentially private** if for every pair of neighbouring datasets $D, D' \in \mathcal{D}$ and every subset $S \subseteq Range(A)$ we have that,

$$Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S] + \delta.$$

That is, the analyst, who we might also view as the adversary, has limited ability to distinguish between these two outputs of the mechanism: $A(D)$ and $A(D')$.

In networks, as well as other forms of datasets, a lot of the information that the analyst hopes to obtain can be expressed in the form of queries. A *query* is a function $f : \mathcal{D} \to \mathbb{R}$, which maps the dataset to a real number. In the case of networks, examples of such questions are minimal cut, specific subgraph counts, maximal independent set, and so on.

Given such queries, we can define *local* and *global* sensitivities as follows: the local sensitivity of a dataset D is $LS_f(D) = \max_{D \sim D'} |f(D) - f(D')|$, and the global sensitivity, which we denote by GS_f, is the maximum local sensitivity over all values $D \in \mathcal{D}$.

It is well known that the Laplace mechanism $A(D) = f(D) + Lap(GS_f/\epsilon)$ preserves $(\epsilon, 0)$-differential privacy [5]. This mechanism is standard for answering queries that have low global sensitivity. This, unfortunately, is not the case for queries involving networks in general. In deed, it is easy to construct instances with high sensitivity by considering special cases of networks where the underlying graph is the empty graph, disconnected (or a graph with a small min-cut), the star graph, the complete graph, or a bipartite graph.

The particular challenge we are grappling with here is that while the local sensitivity might be low, adding noise that is proportional to the local sensitivity invades privacy since it might leak information. One suggestion for how to approach this challenge is smooth sensitivity:

Definition 3 (Smooth sensitivity)**.** A β-smooth upper bound on the local sensitivity of a query f is a function $S_{f,\beta}$ which satisfies the following conditions:

1. $S_{f,\beta}(D) \geq LS_f(D), \forall D \in \mathcal{D}$,

2. $S_{f,\beta},(D) \leq S_{f,\beta}(D') \cdot exp(-\beta d(D,D'))S_{f,\beta}(D'), \forall D, D' \in \mathcal{D}$.

It is possible to preserve privacy while adding noise that is a function of an upper bound on the β-smooth sensitivity of a query. To evaluate such a mechanism, however, one must give an algorithm to efficiently compute the β-smooth sensitivity upper-bound $S_{f,\beta}(G)$. It is not clear that this can be done in polynomial time.

The fourth, and most crucial to this paper, notion of sensitivity that we consider is known as *restricted sensitivity*. This is introduced in [4], partially as a way to use prior knowledge the analyst might have about the networks to address the challenges faced by the other sensitivity concepts. It is reasonable to proceed in this way for social networks because there are various properties (e.g., maximum degree, diameter, etc) that certain social networks have that are common knowledge. We thus take advantage of a hypothesis about our network to restrict the sensitivity of the query. We denote the hypothesis by $\mathcal{H} \subseteq \mathcal{D}$. This hypothesis is true if the true dataset D is an element of $\mathcal{H}$.

Definition 4 (Restricted sensitivity)**.** The restricted sensitivity of f over a hypothesis $\mathcal{H}$ is,

$$RS_f(\mathcal{H}) = max_{\mathcal{D}_1,\mathcal{D}_2 \in \mathcal{H}} \frac{|f(D_1) - f(D_2)|}{d(D_1, D_2)},$$

where the distance is over all paths in $\mathcal{D}$.

We note that $RS_f(\mathcal{H})$ smaller than $LS_f(D)$ and significantly smaller than $S_{f,\beta}(D)$ for some $D \in \mathcal{H}$, for reasons discussed in [4].

4 Results and Discussion

In addressing the issue of private link prediction, our first attempt was to add noise to each score score(x, y) to obtain a new score score$(x, y)'$. We listed the perturbed set in decreasing order of score-value and outputted the top 1000.[2] We were able to do this for all five graphs and experimental results in each case showed that we obtained G_p that is, in various senses, close to G_2. However, this method does not work since it is hard to prove privacy guarantees since the noise that we are adding is local. We could have done experimental results on this to test privacy, but this seems not to be a tractable task given the time-constraints and computational-availability.

Another idea is to use β-smooth sensitivity in order to approximate the notion of local sensitivity, as we already have said in previous sections. The primary constraint here is that it was difficult to find an efficient algorithm for β-smooth sensitivity even for the simple case of common neighbors, despite our extensive attempts. In fact, one can devise faster algorithm that just enumerating over all graphs but this does not beat the exponential dependence on n.

[2] If the list contains less than 1000 scores, then we take the entire list.

4.1 Restricted Sensitivity and Graph-Cut Problems

Since the aforementioned two attempts did not give any results, we switch gears and consider the following scenario: suppose that we have a social network and we want to predict how many Americans are going to collaborate with non-Americans in the second time period. (Alternatively, one can also how many men are going to collaborate with how many women, and so forth.) This corresponds to a specific cut query. Our goal is to predict the edges of the new graph and then release that number such that it does not violate the privacy of any of the nodes. To be more precise, we give the following definition and notations:

Definition 5. We denote by:

- G_1, the graph of the first time-interval. We will call this graph 1-type graph.
- G_p, the graph that occurs after link prediction of the graph G_1. We will call this graph p-type graph.
- G_2, the graph of the second time interval. We will call this graph 2-type graph.
- $cut_G(S)$, the cut on graph G with the one set of the partition being S.

Recall the first of our two notions of neighboring networks.

Definition 6. Two networks G_1 and G_1' are neighboring if and only if they differ on exactly one edge, i.e. the symmetric difference of their edge-sets is exactly a single edge.

Thus, when the analyst makes the query regarding this cut, we run the link prediction algorithm on G_1 and obtain G_p. We then compute the pre-specified cut on G_p. We add a certain amount of Laplace noise to that number so we have $(\epsilon, 0)$ differential private guarantee. To do this, however, we use the notion of restricted sensitivity, defined in the previous section, with the following hypothesis:

Definition 7. The class $\mathcal{H}_k$ is defined as the set $\{G : \forall u, \deg(u) \leq k\}$.

Social networks are often of bounded degree. In this paper, we are looking at the co-authorship graph, and it is reasonable to assume that $k = 20$, i.e., that a single co-author will be involved in at most 20 collaborations over a three year period. Under this assumption, we will make use of the following two very important results from [4]:

Theorem 8. *In the edge-adjacency model, there exists an efficiently computable* 3*-smooth projection* μ *to* $\mathcal{H}_k$*.*

Theorem 9. *Given any query* f *on a social network, the mechanism that uses the projection* μ *from the previous theorem, and answers the query using the mechanism,*

$$A(f, G) = f\left(\mu(G)) + Lap\left(3 \cdot \frac{RS_f(\mathcal{H}_k)}{\epsilon}\right)\right),$$

is $(\epsilon, 0)$*-differentially private.*

We want to release the size of the desired cut on G_p. As indicated in the theorem, we project G_1 and get $\mu(G_1)$. We then run the link prediction algorithm on the projected graph and obtain a score for each edge. We take the top 1000 edges with the highest score and calculate the cut on this predicted graph. We then add Laplace noise proportional to the restricted sensitivity of the function we are using, hence ensuring privacy.

We use two utility functions, neither of which we release, to measure the utility of the combination of the link prediction method and the private mechanism that we are using. We want to release a reasonable estimate of the cut, but we also try to be precise in terms of the original cut and not the predicted graph (i.e., what collaborations actually happened and not what collaborations are estimated in G_p).

Assume that R is the value that we release that estimates a cut. Here, the particular cut that we are considering is S^*. Then, the two utility functions we consider are,

- $u_1 = \frac{R}{|cut_{G_2}(S^*)|}$
- $u_2 = \frac{R - |cut_{G_2}(S^*)|}{R}$

In light of this, we present two main results for the restricted sensitivity of the methods we are using:

Theorem 10. *Under the hypothesis* H_Δ*, the restricted sensitivity using the link predictions methods common neighbors method, preferential attachment, and Jaccard coefficient, is at most* $2(\Delta - 1)$.

Proof. Let D, D' be two datasets that differ on exactly d edges. Imagine starting from D and performing a sequence of events of adding and removing one edge at a time to get to D'. Assume we remove the edge $\{x, y\}$. Let $\Gamma(x)$ be the neighborhood of x and $\Gamma(y)$ the neighborhood of y. After removing this edge, every pair $\{u, x\}$ with $u \in N(y)$ has its score decrease by 1. Likewise, every pair $\{u, y\}$ with $u \in \Gamma(x)$ has its score decrease by 1. The number of such pairs is at most $2\Delta - 2 = 2(\Delta - 1)$. We can use a similar argument for adding an edge.

Therefore, after d steps, at most $(2\Delta - 1)d$ nodes will have their scores altered. Thus, at most $2(\Delta - 1)d$ that were in the cut can get below the threshold, making the maximum change of the cut $(2\Delta - 1)d$. Therefore, the restricted sensitivity is at most $2(\Delta - 1)d/d = 2(\Delta - 1)$. □

Perhaps surprisingly, this bound is actually tight for the common neighbors method. To see this, consider two star graphs of equal sizes. Then, include an edge between their central nodes, and running common neighbors gives us this tight bound.

We now present some experimental results to show the values for Utility 2. This suffices since we can easily obtain values for Utility 1 from the Utility 2 ones. Each category

consists of 100 executions. We choose an arbitrary, and not a random, cut to run this. More specifically, we partition the graph into two equal parts: we sort the authors by time of their first paper in the first time-interval and pick the top half of this list to form one partition and the remaining authors to form the second partition. The second group might roughly correspond to authors that are younger and/or less productive during this time-frame and we want to see the collaborations that emerge with those in the first group in the second time-frame. The values presented in the table below correspond to (mean, variance) of the utility output for the executions corresponding to each cell.

Section	**Common Neighbors**	**Jaccard's Coefficient**	**Pref. Attachment**
astro-ph	63.16, 9.80	111.39, 590.23	335.39, 93.14
cond-mat	17.13, 3.34	31.23, 38.75	94.28, 39.91
gr-gc	4.28, 0.32	11.36, 0.89	63.46, 11.35
hep-ph	14.45, 2.54	27.46, 28.97	12.59, 20.65
hep-th	10.83, 1.85	14.21, 2.16	0.34, 0.18

The histogram below presents similar data. The y-axis denotes the number of executions and the x-axis gives the utility value. The thickness of the histogram is to denote the range of utility values that are accounted for by the bar.

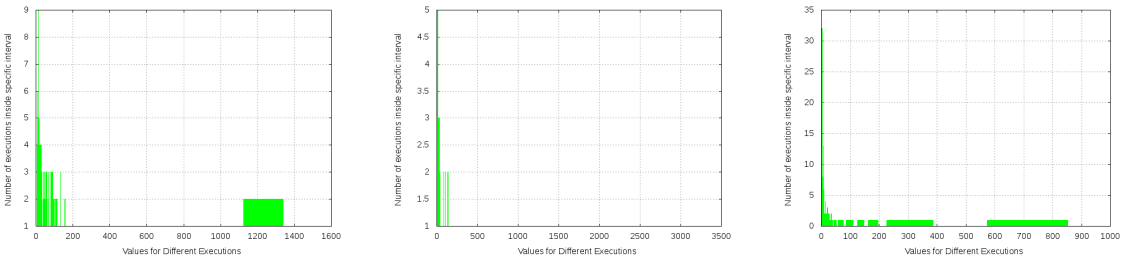

Figure 2: Astrophysics

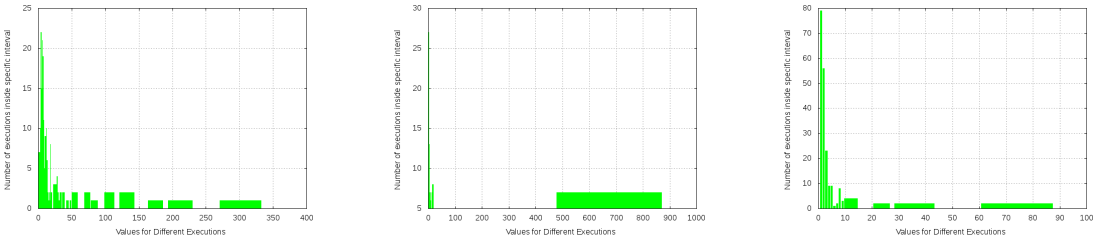

Figure 3: Condensed Matter Physics

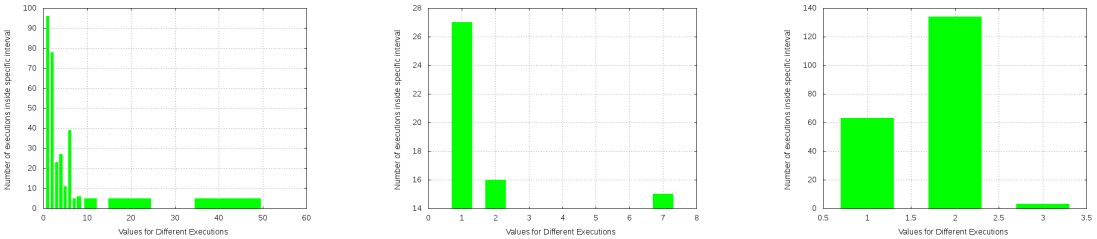

Figure 4: General Relativity and Quantum Cosmology

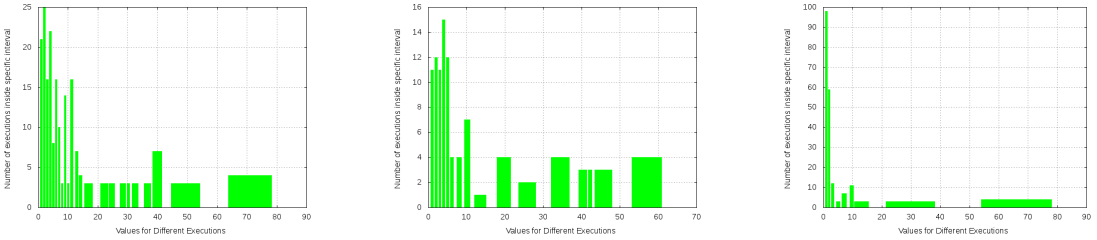

Figure 5: High Energy Physics-Phenomenology

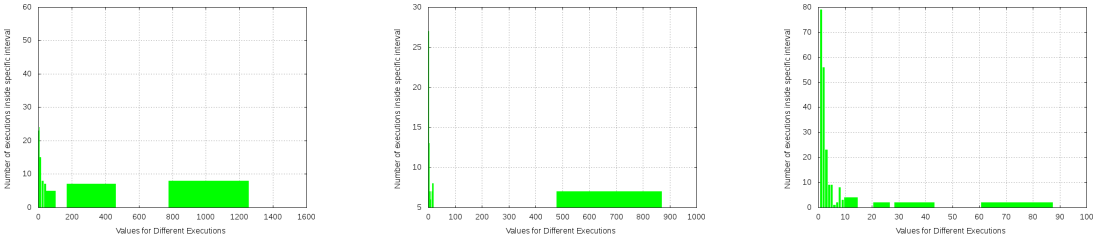

Figure 6: High Energy Physics-Theory

We observe that even under our hypothesis, the noise we add is quite big. If the output is not accordingly big enough, we expect to have large variance and expectation that are almost the expectation and variance of the Laplace noise that we added. In the next section, we consider a different utility function in order to bypass this issue. From our diagrams, we also observe that from the three methods, there are no clear winners, although Jaccard's coefficient and common neighbors seem to perform slightly better than preferential attachment in general. One potential direction hinted by this is to come up with a better bound for preferential attachment in order to have better performance. Another, more ambitious, direction is to embed randomness inside the link prediction algorithm in order to maintain privacy and check how the noise affects the prediction.

4.2 Further Attempts Using Restricted Sensitivity

We consider another attempt using the second notion of neighboring datasets. Recall that in the former case we were only considering graphs from the training interval. Here, we let our database be a pair of graphs (G_1, G_2). We then have the following definitions:

Definition 11. Two databases $D = (G_1, G_2)$ and $D' = (G_1', G_2')$ are said to be neighboring if and only if $G_1 \sim G_1'$ and $G_2 \sim G_2'$.

Definition 12. Let S and T be two sets. Then the set difference of S and T is,

$$\mathcal{SD}(S,T) = \{x : x \in (S - T) \cap (T - S)\}.$$

We want to release the value $|\mathcal{SD}(\mathrm{cut}(G_p), \mathrm{cut}(G_2))| - |\,\mathrm{cut}(G_2)|$, under this definition. This comes in handy since it not only takes that the we got the size of the cut correct but that we were also taking into account the right set of edges. As before, we will use Laplace noise on the output to guarantee privacy. We will also present some experimental results.

We proceed similar to what we did before: project G_1 using the theorem above, run the link-prediction algorithm and take the top 1000 nodes according to score value (with Laplace noise), compute the symmetric difference above. We give the following privacy guarantee under the maximum degree hypothesis $\mathcal{H}_\Delta$.

Theorem 13. *Under the hypothesis $\mathcal{H}_\Delta$, the restricted sensitivity using any of the three link prediction methods in consideration is at most 2Δ.*

Proof. Take two datasets $D = (G_1, G_2)$ and $D' = (G_1', G_2')$ such that $d(D, D') = d$. Here, this means that the $\max(d(G_1, G_1'), d(G_2, G_2')) = d$. By the triangle inequality, we have to argue how much the symmetric difference and the cut of the 2-type graph changes in this setting. The second term is easy since it can only change by at most d. For the first term, a similar argument as in Theorem 10 will show that it can change by at most $(2\Delta - 1)d$. This is because each edge deletion or insertion in both graphs can change the difference by at most $(2\Delta - 1)$. This is one more than before since we are now considering neighboring datasets that differ by a single edge not only on their 1-type graph but also their 2-type graph. □

We are able to provide strong utility guarantees for graphs in both time-frames. Recall that we are not releasing the utility and we do not give privacy guarantees related to how to release that privately. But, we run experiments to give a ballpark of how these methods perform. For each of the results below, we run 100 experiments on each of the link prediction methods. In this subsection, as our database contains both graphs G_1 and G_2, the threshold in the link prediction method is defined as in the Kleinberg paper: we take the X edges with the highest score, where $X = |E_{G_2} \cap \mathrm{core} \times \mathrm{core}\,|$. The table gives the corresponding values (mean, variance) for each cell.

Section	Common Neighbors	Jaccard's Coefficient	Pref. Attachment
astro-ph	1285.36, 8260.68	1271.87, 8088.32	261263.86, 7986.73
cond-mat	449.16, 1008.74	434.37, 943.41	459.48, 1055.63
gr-gc	94.73, 44.87	104.82, 54.93	136.59, 93.28
hep-ph	224.51, 252.02	257.82, 332.37	239.29, 286.31
hep-th	187.31, 175.43	229.22, 262.70	189.63, 179.80

The histograms below give the same data. The y-axis denotes the number of executions and the x-axis gives the utility value. The thickness of the histogram is to denote the range of utility values that are accounted for by the bar. Therefore, for instance, the most popular value for common neighbors for the astrophysics graph is around 3000.

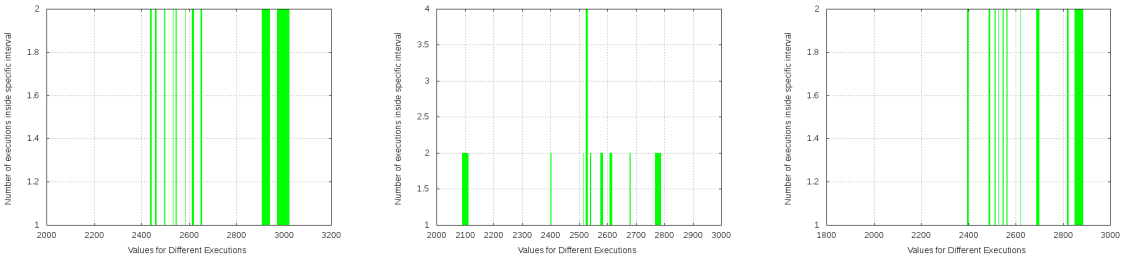

Figure 7: Astrophysics

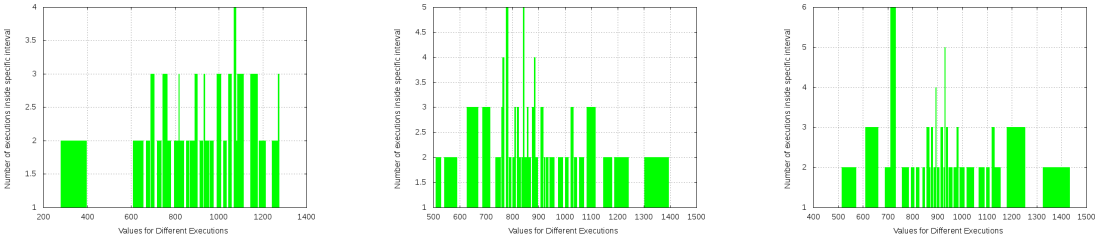

Figure 8: Condensed Matter Physics

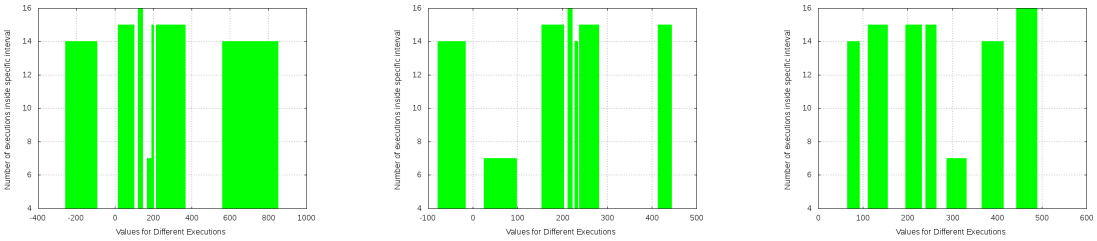

Figure 9: General Relativity and Quantum Cosmology

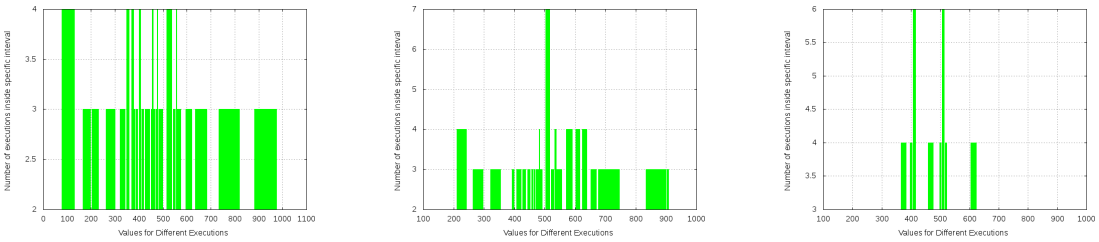

Figure 10: High Energy Physics-Phenomenology

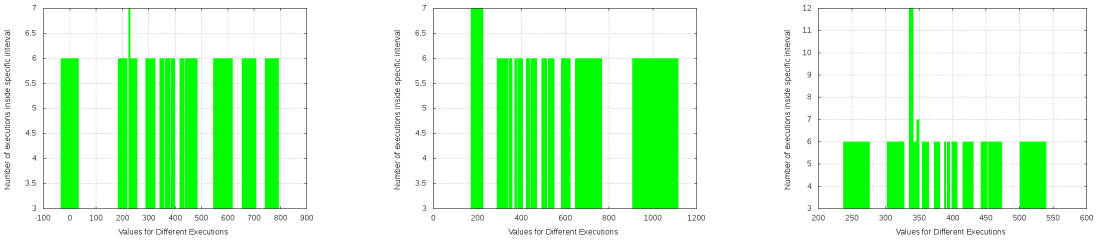

Figure 11: High Energy Physics-Theory

The definition of new function tries to capture the relevance between the 1-type graph and the 2-type graph, or equivalently how good the link prediction method was, so we are trying to release this value in a differentiable private way. Also, since the distribution of the Laplace noise we are adding is almost the same in both this and the previous subsection and the value we are trying to release is less negligible than before, we expect our experiments to be more accurate.

5 Conclusion and Further Directions

Privacy on networks is a very hot topic, but very little work has been done on it due to the difficulty nature of the questions. This project attempts to tackle one aspect of this field, but in the process raises several more. There are several directions that we would like to pursue in the future. One clear direction is improving the restricted sensitivity bounds for Jaccard's coefficient and preferential attachment (or show that these bounds are tight).

In this project, it was easier for us to define a specific cut problem and answer queries about that cut in a differentially private way by adding noise to the final cut-size. Another attempt, which we have mentioned earlier and we think is much more ambitious, is to embed randomness inside the link prediction algorithm in order to maintain privacy. We can then check how much the noise affects the predictions and see if we are able to get better utility guarantees. This would allow us not to treat these methods as a black box but instead take advantage of the knowledge of these very well-known methods.

It would also be useful to generalize our results not just to cut queries, but other questions regarding the networks including diameter, degree distribution, min-cut, and so on to see whether we are able to give any privacy guarantees in this context.

References

[1] Mohammad Al Hasan and Mohammed J. Zaki. A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. Springer US, 2011.

[2] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[3] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: Improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 61–70, New York, NY, USA, 2010. ACM.

[4] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. *CoRR*, abs/1208.4586, 2012.

[5] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy.* Foundations and Trends in Theoretical Computer Science. Now Publishers Incorporated, 2014.

[6] David Easely and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, New York, NY, USA, 2010.

[7] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *ACM Transactions on Database Systems*, 39(3):22:1–22:33, October 2014.

[8] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science Technology*, 58(7):1019–1031, May 2007.

[9] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: Inferring user profiles in online social networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 251–260, New York, NY, USA, 2010. ACM.

[10] Mark Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[11] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 75–84, New York, NY, USA, 2007. ACM.

[12] President's Council of Advisors on Science and Technology. Big data and privacy: A technological perspective. Technical report, Executive Office of the President, 5 2014.

[13] Jie Tang, Tiancheng Lou, and Jon Kleinberg. Inferring social ties across heterogenous networks. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 743–752, New York, NY, USA, 2012. ACM.