Preserving the Privacy of Sensitive Relationships in Graph Data

Elena Zheleva and Lise Getoor

Computer Science Department
University of Maryland
College Park, MD
{elena,getoor}@cs.umd.edu

Abstract. In this paper, we focus on the problem of preserving the privacy of sensitive relationships in graph data. We refer to the problem of inferring sensitive relationships from anonymized graph data as *link re-identification*. We propose five different privacy preservation strategies, which vary in terms of the amount of data removed (and hence their utility) and the amount of privacy preserved. We assume the adversary has an accurate predictive model for links, and we show experimentally the success of different link re-identification strategies under varying structural characteristics of the data.

Keywords: privacy, anonymization, identification, link mining, social network analysis, noisy-or, graph data

1 Introduction

The goal of data mining is discovering new and useful knowledge from data. Sometimes, the data contains sensitive information, and it needs to be sanitized before it is given to data mining researchers and the public in order to address privacy concerns. Data sanitization is a complex problem in which hiding private information trades off with utility reduction. The goal of sanitization is to remove or change the attributes of the data which help an adversary infer sensitive information. The solution depends on the properties of the data and the notions of privacy and utility in the data.

Most of the work in this area makes the assumption that the data is described by a single table with attribute information for each of the entries. However, real-world datasets often exhibit more complexity. Relational data, often represented as a multi-graph, can exhibit rich dependencies between entities. The challenge of anonymizing graph data lies in understanding these dependencies and removing sensitive information which can be inferred by direct or indirect means.

Very little work has been done in this direction, and there has been a growing interest in it. The existing work looks at the identifying structural properties of the graph nodes [2, 7], or considers relations to be attributes of nodes [13]. Our work assumes that the anonymized data will be useful only if it contains both

structural properties and node attributes. We study anonymization techniques to match this assumption.

Another distinction of our approach is that, unlike existing work on privacy preservation which concentrates on hiding the identity of entities, we look at the case where relationships between entities are to be kept private. Finding out about the existence of these sensitive relationships leads to a privacy breach. We refer to the problem of inferring sensitive relationships from anonymized graph data as *link re-identification*.

Examples of sensitive relationships can be found in social networks, communication data, search engine data, disease data and others. In social network data, based on the friendship relationships of a person and the public preferences of the friends such as political affiliation, it may be possible to infer the personal preferences of the person in question as well. In cell phone communication data, finding that an unknown individual has made phone calls to a cell phone number of a known organization can compromise the identity of the unknown individual. Another example is in search data: being able to link search queries made by the same individual can give personal information that helps identify that individual. In hereditary disease data, knowing the family relationships between individuals who have been diagnosed with hereditary diseases and ones that have not, can help infer the probability of the healthy individuals to develop these diseases.

We consider the node data to be anonymized using a known single-table definition such as k-anonymization [16] or the more recently proposed t-closeness [8]. For the edge data, we propose five different anonymization strategies. The most conservative approach is to remove the relationships altogether, thus preserving any privacy that these relationships may compromise. We assume that while all of the sensitive relationships are removed, all or a portion of the relationships of other types are left intact in the anonymized data. We propose a method which allows modeling the influence of data attributes on sensitive relationships, and studying how different anonymization techniques can preserve privacy. The privacy breach is measured by counting the number of sensitive relationships that can be inferred from the anonymized data. The utility of the data is measured by counting how many attributes or observations have to be deleted in the sanitization process.

To formalize privacy preservation, Chawla et al. [4] propose a framework based on the intuitive definition that "our privacy is protected to the extent we blend in the crowd." What needs to be specified in this general framework is an abstraction of the concept of a database, the adversary information and its functionality, and when an adversary succeeds. Starting from this idea, we define the relational privacy framework for link re-identification. After the background overview in Section 2, we define the data model in Section 3. We then discuss methods for anonymizing graph data and the resulting adversary information in Section 4. Section 5 covers graph-based privacy attacks, Section 6 discusses general link re-identification attacks, and Section 7 discusses link re-identification in anonymized data and when an adversary succeeds. Section 8 presents the

benefits and disadvantages of each anonymization method in an experimental setting, and Section 9 contains concluding remarks and ideas for future work.

2 Background and Related Work

Until recently, the literature on privacy preservation considered the data to be a single table, in which the rows represent records, and the columns represent attributes [1, 3, 8, 9, 12, 18]. However, real-world data is often relational, and records may be related to one another or to records from other tables. For example, a database for studying hereditary diseases can contain both patient medical records and family relationships between patients. A database for studying the social network structure in a university department can contain both student information together with enrollment and research group data. Another example is data for studying Internet traffic, in which the sequences of packet traces are related to each other [14].

It is well known that even in single-table data, removing the identifying information such as social security number is not enough for preserving the privacy of individuals represented in data [18]. One of the most popular techniques for anonymizing single table data is k-anonymity, in which the quasi-identifying attributes of the table records are altered in a way that each record becomes indistinguishable from at least $k-1$ other records [16]. The set of records with the same anonymized attributes forms an equivalence class. Since k-anonymity was first introduced, various methods for k-anonymizing data have been developed in the research community [1, 3]. Recently proposed anonymity definitions such as l-diversity [9] and t-closeness [8] address some of the deficiencies of k-anonymity. l-diversity addresses the concern that an equivalence class may not contain diverse enough sensitive attributes. t-closeness addresses the stronger concern that the distribution of sensitive attributes in an equivalence class may not match the distribution of sensitive attributes in the whole data set. More definitions of privacy and information disclosure can be found in [4, 5, 10, 11].

While it is possible to represent the nodes of a graph in a single table if the nodes have the same type, it is not clear how to do that when the nodes exhibit relationships and when there are nodes of different types. Very little work has been done on privacy preservation in graph data. Only recently, there has been privacy research on identifying structural properties of graph nodes [2, 7], or on applying k-anonymity to multi-relational data [13]. The model of Miklau et al. [7] defines k-candidate anonymity for graph data based on the degrees of the nodes in the neighborhoods of the nodes to be anonymized. Their experiments on real-world datasets show that the more someone knows about the neighborhood of a node, the higher the probability for this node to be identified uniquely. They create an approach for anonymizing structure by random deletion and addition of edges. Their model assumes that the nodes and edges do not contain any attributes besides a random identifier; here, we consider models with attributes and links.

Similarly, Backstrom et al. [2] consider graphs in which the structural properties of the anonymized nodes can help an adversary to find the real-world entities behind these nodes. They consider social networks in which the node attributes are stripped off, and the edges are kept intact. They describe two families of attacks on the privacy of communication in these networks: active and passive attacks. In the active attacks, the adversary "inserts" himself in the network by creating connections with people of interest, and then tries to find himself in the anonymized version. These attacks assume that the owner of the data releases the full graph data periodically. The passive attacks assume that the adversary and his colluding friends can identify themselves in the network.

Nergiz and Clifton [13] recognize the problem that existing k-anonymizing approaches apply only to single-table data, and they extend k-anonymity to apply to relational data. Their approach abstracts the knowledge about a private entity from multiple tables into a k-anonymized tree. It keeps relationships between entities of different types but it does not discuss relationships between the entities whose privacy is a concern. Not keeping such relationships would remove some of the structural properties which are interesting in graph data.

Privacy preservation in graph data is closely related to link mining. Graph data exhibits dependencies, and they can be used to learn about identities, classes and relationships represented in it. They have been studied in the link mining community [6], and the techniques developed for collective classification, object identification and link prediction can be used to learn hidden properties of the data. If these hidden properties are sensitive, then there is a privacy breach. In this paper, we are mostly concerned with link prediction. Link prediction uses properties of the graph in order to determine whether two nodes in the graph exhibit a relationship of a particular type. For example, it may predict whether two people in a social network graph are likely to be friends. The knowledge that two people have many opportunities for communication makes them more likely to be friends, and it can be exploited by an adversary to predict likely friendships.

3 Data Model

We consider graph data which describes entities and relationships between entities. We assume that the relationships are binary relationships. In a graph, entities are represented by nodes, and relationships by edges. In general, we can have different types of nodes and different types of edges. For the purposes of this paper, we focus on the case where there is a single node type and multiple edge types. We distinguish one of the relationship types as the *sensitive relationship*. This is the relationship which we are interested to hide from the adversary. The nodes and edges can have associated attributes. In addition, the graph has structural properties. Structural properties of a node include node degree and neighborhood structure.

More formally, we consider a database describing a multi-graph $G = (V, E^1, \ldots, E^k, E^s)$, composed of a set of nodes V and sets of edges $E^1, \ldots, E^k, E^s$.

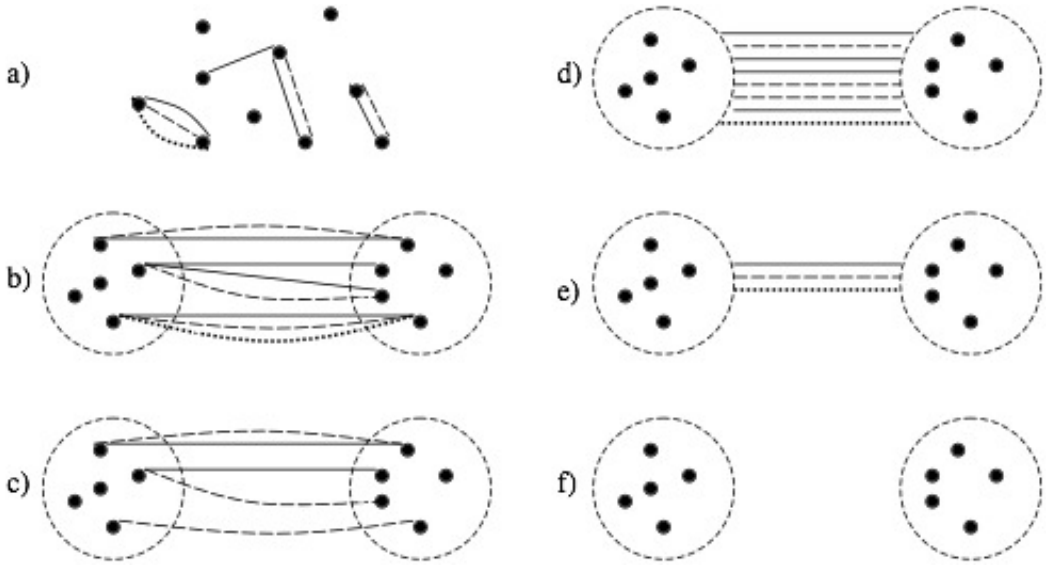

Fig. 1. The original data graph (a)) and the output from five anonymization approaches to graph data: b) revealing the observations between nodes, c) removing 50% of the observations , d) revealing all the observations between equivalence classes of nodes (cluster-edge anonymization), e) constrained revealing of the observations between equivalence classes of nodes (cluster-edge anonymization with constraints), f) removing all relational observations. There are three different edge types in the original data graph represented by different line styles. Clusters resulting from node anonymization are circled with dotted lines.

Each node v_i represents an entity of interest. An edge $e^1_{i,j}$ represents a relationship of type E^1 between two nodes v_i and v_j. The $E^1, \ldots, E^k$ are the *observed* relationships, and E^s is the sensitive relationship, meaning that it is undesirable to disclose the e^s edges to the adversary.

In the process of anonymizing the data, the sensitive relationships are always removed, i.e., they are not provided in the released data. However, it may be possible to predict some of these relationships using other observed relationships and/or node attributes. For the purposes of this paper, we focus on predicting sensitive edges based on the observed edges, but it is straightforward to include node and edge attributes and interesting to also consider structural properties. If the sensitive edges can be identified, then we say that there has been a *privacy breach*.

In addition, the data can include certain *constraints* which specify the number of relationships of a particular type or the number of relationships connecting any two nodes. Constraints can also be inequality constraints describing the maximum or minimum number of relationships.

As a motivating example, consider the case where the entities are students, and the relationships between students v_i and v_j include taking a class c together (`classmates`(v_i, v_j, c)), belonging to the same research group (`groupmates`(v_i, v_j, g)), and being friends (`friends`(v_i, v_j)). We can consider the class and research groups as attributes of the edges, so that students can take more than one class together, and they can belong to more than one research group. In this case, we may consider `friends` to be the sensitive relationship. We are interested in understanding how difficult it is to determine friendship based on class and research group rosters.

4 Graph Anonymization

The process of anonymization involves taking the unanonymized graph data, making some modifications, and constructing a new *released graph* which will be made available to the adversary. The modifications include changes to both the nodes and edges of the graph. We discuss several graph anonymization strategies and, for each approach, we discuss the tradeoffs between privacy preservation and the utility of the anonymized data.

We assume that the adversary has the information contained in the released graph data, and the constraints on the data. The *adversary succeeds* when she can figure out whether two nodes exhibit a sensitive relationship, i.e., when she is able to correctly predict a sensitive link between them. For example, if the adversary can figure out which students are likely to be friends given the released graph, then the data discloses private information about the two individuals.

4.1 Node anonymization

We assume that the nodes have been anonymized with one of the techniques introduced for single table data. For example, the nodes could be k-anonymized using t-closeness [8]. This anonymization provides a clustering of the nodes into m equivalence classes $(C_1, \ldots, C_m)$ such that each node is indistinguishable in its quasi-identifying attributes from some minimum number of other nodes. We use the following notation $C(v_i) = C_k$ to specify that a node v_i belongs to equivalence class C_k.

The anonymization of nodes creates equivalent classes of nodes. Note, however, that these equivalent classes are based on node attributes only, and inside each equivalence class, there may be nodes with different identifying structural properties and edges.

4.2 Edge anonymization

For the relational part of the graph, we describe five possible anonymization approaches. They range from one which removes the least amount of information to a very restrictive one, which removes the greatest amount of relational data. Figure 1(a) shows a simple data graph in which there are ten nodes and eight

observed edges. There are three edge types, and each one is represented by a different line style. We will illustrate each of our techniques on this graph. For each approach, we discuss the tradeoffs between privacy preservation and the utility of the anonymized data.

Intact edges The first (trivial) edge anonymization option is to only remove the sensitive edges, leaving all other observational edges intact. Figure 1(b) shows an illustration of this technique applied to the original data graph of Figure 1(a).

In our running example, we remove the friendship relationships, since they are the sensitive relationships, but we leave intact the information about students taking classes together and being members of the same research group. Since the relational observations remain in the graph, this anonymization technique should have a high utility. But it is likely to have low privacy preservation.

Intact-Edge Anonymization Algorithm

1: Input: $G = (V, E^1, \ldots, E^s)$
2: Output: $G' = (V', E^{1'}, \ldots, E^{k'})$
3: V'=anonymize-nodes(V)
4: **for** t=1 to k **do**
5: $\quad E^{t'} = E^t$
6: **end for**

Fig. 2. Algorithm for anonymizing graph data by removing only the sensitive edges.

Partial-edge removal Another anonymization option is to remove some portion of the relational observations. We could either remove a particular type of observation which contributes to the overall likelihood of a sensitive relationship, or remove a certain percentage of observations that meet some pre-specified criteria (e.g., at random, connecting high-degree nodes, etc.). Figure 1(c) shows an illustration of this technique when the edges are removed at random.

This partial edge removal process should increase the privacy preservation and reduce the utility of the data as compared to the previous method. Removing observations should reduce the number of node pairs with highly likely sensitive relationships but it does not remove them completely. For those pairs of nodes, private information may be disclosed.

Cluster-edge anonymization In the above approaches, while the nodes had been anonymized, the number of nodes in the graph was still the same, and the edges were essentially between copies of the anonymized nodes. Another approach is to collapse the anonymized nodes into a single node for each cluster, and then consider which edges to include in the collapsed graph.

The simplest approach is to leave the sets of edges intact, and maintain the counts of the number of edges between the clusters for each edge type. We refer to

Partial-Edge Anonymization Algorithm

Input: $G = (V, E^1, \ldots, E^k, E^s)$, percent-removed
Output: $G' = (V', E^{1'}, \ldots, E^{k'}))$
V'=anonymize-nodes(V)
for t=1 to k **do**
  $E^{t'} = E^t$
  removed = $\lceil$percent-removed $\times \|E^{t'}\| \rceil$
  for i=1 to removed **do**
    e_i = random edge from $E^{t'}$
    $E^{t'} = E^{t'} \setminus \{e_i\}$
  end for
end for

Fig. 3. Algorithm for anonymizing graph data by removing randomly a portion of the observed edges.

this technique as *cluster-edge* anonymization. Figure 4 presents the algorithm for this technique, and Figure 1(d) shows an illustration of the result from applying the algorithm.

Cluster-Edge Anonymization Algorithm

Input: $G = (V, E^1, \ldots, E^k, E^s)$,
Output: $G' = (V', E^{1'}, \ldots, E^{k'}))$
$V' = \{C_1, \ldots, C_m\}$
for t=1 to k **do**
  $E^{t'} = \emptyset$
  for all $(v_i, v_j) \in E^t$ **do**
    $C_i = C(v_i)$
    $C_j = C(v_j)$
    $E^{t'} = E^{t'} \cup \{(C_i, C_j)\}$
  end for
end for

Fig. 4. Algorithm for cluster-edge anonymization technique.

Cluster-edge anonymization with constraints Next, we consider using a stricter method for sanitizing observed edges than the previous technique. The *cluster-edge anonymization with constraints* technique creates edges between equivalence classes as above, but it requires the equivalence class nodes to have the same constraints as any two nodes in the original data. For example, if there can be at most two edges of a certain type between entities, there can be at most two edges of a certain type between the cluster nodes. This, in effect, removes some of the count information that is revealed in the previous anonymization technique.

Cluster-Edge Anonymization with Constraints Algorithm

```
1: Input: G = (V, E)
2: Output: G' = (V', E')
3: V' = {C_1, ..., C_m}
4: for t=1 to k do
5:    E^t' = ∅
6:    for all (v_i, v_j) ∈ E^t do
7:       C_i = C(v_i)
8:       C_j = C(v_j)
9:       if (C_i, C_j) ∉ E^t' then
10:         E^t' = E^t' ∪ {(C_i, C_j)}
11:      end if
12:   end for
13: end for
```

Fig. 5. Algorithm for cluster-edge with constraints anonymization technique.

In order to determine the number of edges of a particular type connecting two equivalence classes, the anonymization algorithm picks the maximum of the number of edges of that type between any two nodes in the original graph. In our earlier example, if the maximum number of common classes that any pair of students from the two equivalence classes takes is one class together, then the equivalence classes are connected by one class edge. Figure 1(e) shows an illustration of this technique.

This information will keep some of the utility of the data but it will say nothing of the distribution of observations. The anonymized data hides whether all observations appear on one two-node edge or on all two-node edges, and whether they ever appear in the same two-node edge. This may reduce the privacy breach on each sensitive relationship.

Removed edges The most conservative anonymization option is to remove all the edges. Depending on the intended uses of an anonymized social network, removing the node and/or edge attributes completely may be undesirable. For example, if one wants to know whether any first-year students took a particular course together, then all the three types of information, i.e., edges, edge attributes and node attributes, are necessary. In our toy example, while taking a course together is information contained in a network edge, the name of the course is an edge attribute, and the year of enrollment is a node attribute. In this case, this anonymization technique would lead to very low utility, yet high privacy preservation.

5 Graph-based Privacy Attacks

According to Li et. al. [8], there are two types of privacy attacks in data: *identity disclosure* and *attribute disclosure*. In graph data, there is a third type of attack:

No-Edge Anonymization Algorithm

```
1: Input: G=(V,E)
2: Output: G'=(V',∅)
3: V'=anonymize-nodes(V)
```

Fig. 6. Algorithm for anonymizing graph data by removing the edges

link re-identification. Identity disclosure occurs when the adversary is able to determine the mapping from an anonymized record to a specific real-world entity (e.g. an individual). Attribute disclosure occurs when the adversary is able to infer the attributes of a real-world entity more accurately than it would be possible before the data release. Identity disclosure often leads to attribute disclosure [8]. Both identity disclosure and attribute disclosure have been studied very widely in the privacy community [1–4, 7–9, 11–13, 16, 18].

Rather than focus on these two kinds of attack, the focus of our paper is on link re-identification. Link re-identification is the problem of inferring that two entities participate in a particular type of sensitive relationship or communication. *Sensitive conclusions* are more general statements that an adversary can make about the data, and can involve both node, edge and structural information. These conclusions can be the results of aggregate queries. For example, in a database describing medical data informal about company employees, finding that almost all people who work for a particular company have a drinking problem may be undesirable. Depending on the representation of the data, this can be revealed by using both the node attributes and the co-worker relationship.

6 Link Re-identification Attacks

The extent of a privacy breach is often determined by data domain knowledge of the adversary. The domain knowledge can influence accurate inference in subtle ways. The goal of the adversary is to determine whether a sensitive relationship exists. There are different types of information that can be used to infer a sensitive relationship: node attributes, edge existence, and structural properties. Based on the domain knowledge of the adversary, she can construct rules for finding likely sensitive relationships. In this work, we assume that the adversary has an accurate probabilistic model for link prediction, which we will describe below.

In our running example, the sensitive friendship link may be re-identified based on node attributes, edge existence or structural properties. For example, consider two student nodes containing a boolean attribute "Talkative." Two nodes that both have it set to "true" may be more likely to be friends than two nodes that both have it set to "false." This inference is based on node attributes. An example of re-identification based on edge existence is two students in the same research group who are more likely to be friends compared to if they are in different research groups. A re-identification that is based on a structural property such as node degree would say that two students are more likely to be

friends if they are likely to correspond to high degree nodes in the graph. A more complex observation is one which uses the result of an inferred relationship. For example, if each of two students is highly likely to be a friend with a third person based on other observations, then the two students are more likely to be friends too.

6.1 Link re-identification using observations

We assume that the adversary has a probabilistic model for predicting the existence of a sensitive edge based on a set of observations $\boldsymbol{O}$: $P(e^s_{ij}|\boldsymbol{O})$. In this work, we assume a simple *noisy-or model* [15] for the existence of the sensitive edge. The noisy-or model can capture the fact that each observed edge contributes (in a probabilistic way) to the probability of the sensitive edge existing; it makes the simplifying assumption that each factor is an independent cause for the sensitive edge. Here, we focus on re-identification based on edge existence, so the observations that we consider are sets of edges, e^l_{ij}. For simplicity, we label these observations $o_1, \ldots, o_n$. For each observed edge, we assume that we have a *noise* parameter, $\lambda_1, \ldots, \lambda_n$, and, in addition, we have a *leak* parameter λ_0 which captures the probability that the sensitive edge is there due to other, unmodeled, reasons. A noise parameter λ_i captures the independent influence of an observed relationship o_i on the existence of a sensitive relationship. Then, according to the noisy-or model, the probability of a sensitive edge is:

$$P(e^s_{ij} = 1) = P(e^s_{ij} = 1|o_1, ..., o_n) = 1 - \prod_{l=0}^{n}(1 - \lambda_l)$$

The above formula applies only when the observations are certain. It is also possible that the observation existence is not known. In that case, there are probabilities $P(o_1), \ldots, P(o_n)$ associated with the existence of each observation, and the probability of a sensitive edge is:

$$P(e^s_{ij} = 1) = \sum_{\{\mathbf{o}\}} P(e^s_{ij} = 1|\mathbf{o}) \prod_{k=1}^{n} P(o_k)$$

where

$$P(e^s_{ij} = 1|\mathbf{o}) = 1 - (1 - \lambda_0) \prod_{l=1}^{n}(1 - \lambda_l)^{o_l}$$

More details about this model can be found in [17].

The noisy-or function is applicable when there are a few observations that can cause an event, and each one can contribute positively to the likelihood of the event, independent of the rest. The function has some nice properties: 1) the

result of it is always between 0 and 1 when the input probabilities are in that range; 2) the final result is independent of the order in which the observations are added; 3) it can accommodate different number of observations; 4) adding a new positive observation always increases the overall likelihood. We use this function to measure how likely each sensitive relationship is, and to find whether there are parts of the graph that are vulnerable to an adversary attack. It is also possible to express the dependence between events in an explicit probability model such as a Bayesian or a Markov network, when the dependences between observations are known.

6.2 Amount of information disclosed

Based on the noisy-or model for each pair of nodes, it is possible to determine the number of node pairs that are likely to participate in a sensitive relationship. In the anonymized data, it is desirable to have few sensitive relationships which can be inferred with high likelihood. To formalize this desirable property, we can compute the percentage of all possible two-node relationships which have a high likelihood and make sure that it is below some allowed level δ:

$$\frac{|relationships(P(e_{ij}^s) > \rho)|}{|V|^2} < \delta \tag{1}$$

where ρ is the threshold for predicting that a sensitive relationship exists and $relationships(P(e_{ij}^s) > \rho)$ returns the set of all sensitive relationships which have likelihood above ρ. For example, if it is true for the given data that 15% of the possible pair relationships have a true likelihood of exhibiting a sensitive relationship higher than 0.8, then

$$\frac{|relationships(P(e_{ij}^s) > 0.8)|}{|V|^2} <= 0.15.$$

For each anonymization technique, it is possible to find the highest possible δ that satisfies a particular ρ level. This can be used to compare the privacy preservation for each technique. The higher the δ, the lower the privacy preservation.

6.3 Utility

Utility in the data is hard to measure, and we make an assumption that the more observations there are in the anonymized data, the better. To measure utility, we use a very simple approach. We count the number of observations which were removed in the process of anonymization. The lower the number of removed observations, the higher the overall utility. For the intact edge and the cluster-edge anonymization techniques, no relational observations are deleted, therefore, these two techniques have the highest utility. For the partial edge removal technique, the utility depends on the percentage of edges removed. For the cluster-based with constraints technique, it is much lower, since the graph is collapsed, and many edges are removed. The exact number can be computed

using the properties and constraints of the data such as number of nodes, edges of each type, and the size of the equivalence classes. Note that a more sophisticated measure of utility would also consider the loss of structural properties in the anonymized data. In the case when all the edges are removed, the utility is 0.

7 Link Re-identification in Anonymized Data

In the first two types of link anonymization (intact and partial), the noisy-or model can be used directly to compute the probability of a sensitive edge. In the other two cases, one has to consider the probability that an observed edge exists between two nodes, and apply the noisy-or.

7.1 Link re-identification in cluster-edge anonymization

In the case of keeping edges between equivalence classes, the probability of an observation existing between two nodes is not given and it needs to be estimated. The noisy-or function will need to take into consideration the probability associated with each observation in order to compute the likelihood of a sensitive relationship. When the number of relationships of each type (e.g., course, research group, etc.) between two equivalence classes is given, the distribution is not uniform, and the probability of an observation *P(o)=P(observation*(v_i, v_j)*)* existing between two students can be computed directly from the counts of relationships between their equivalence classes. *P(*`classmates(`v_i, v_j, c`)`*)* expresses the probability that there exists a class edge between any two students v_i and v_j from two equivalence classes $C(v_i)$ and $C(v_j)$, i.e., the students take a course c together. It is equal to the number of possible student pairs from the two equivalence classes who take a course together —`classmates(`$C(v_i), C(v_j)$`)`— as a fraction of the number of possible relationships in the graph $|V|^2$.

7.2 Link re-identification in cluster-edge anonymization with constraints

In the constrained cluster-edge anonymization approach, the number of relationships between equivalence classes is not given. Therefore, the probability of an observation existing between any two edges has to be taken into account in the noisy-or model. To estimate this probability, an adversary can assume a uniform distribution, meaning that the probability of an observation existing between any two edges is the same for all edges in the graph. This estimate is worse than the cluster-edge anonymization method. Using the constraints on the data, it is possible to get estimates of this probability. For example, if it is known that there are 50 pairs of students who take courses together, and there are 100 possible pairs, then the probability of any two students taking any class c together is *P(*`classmates(`v_i, v_j, c`)`*)=0.5*. If the adversary knows the number of

offered courses c, the number of courses per person n, the number of students $s = |V|$, and assumes that all courses have the same number of people $p = \frac{s*n}{c}$, then the number of possible pairs who take courses together can be calculated as $n * (p - 1)$. This number can be used to compute in a manner similar to the cluster-edge anonymization method $P($`classmates`$(v_i, v_j, c)) = \frac{n*(p-1)}{|V|^2}$.

One can also use an expected value of any two-node relationship to be sensitive by looking at the likelihood distribution of all relationships. However, we found that this does not measure privacy well because an adversary is more interested in the highly likely relationships.

An observation probability shows the percentage of edges between two nodes from two different equivalence classes that contain the observation. For example, if the two equivalence classes have exactly 10 nodes each, and the observation exists for 30 of the two-node edges, then the edge probability is $P($`observation`$(v_i, v_j)) = 0.3$ where `observation`(v_i, v_j) is either `classmates`(v_i, v_j, c), or `groupmates`(v_i, v_j, g) for any c and g. This increases the utility of the data as compared to the case when no probabilities are included, but it can also decrease the privacy preservation. An exception is the case when observations between equivalence classes have exactly the same distribution as the overall uniform distribution.

8 Experiments

The effectiveness of the anonymization approaches depends on the structural and statistical characteristics of the underlying graph. In order to study the influence of each anonymization approach on privacy preservation, we apply them to synthetic data generated under varying statistical and structural assumptions and compute the information disclosed. We show how many relationships are revealed at different probability thresholds. First, we describe the data generator.

8.1 Data generator

The data generator creates data according to the data model described in Section 3. The input to the data generator includes: the number of nodes, maximum number of nodes which can participate in a relationship (e.g., the maximum number of students taking the same class), the maximum number of relationships that each student can have with any other student (e.g., maximum number of classes that a student can take). For all observation types, the probability of two nodes exhibiting a sensitive relationship given the observation type is given and the leak probability, the probability of two nodes exhibiting a sensitive relationship due to unobserved causes.

For the concrete example, the data generator starts by creating a set of students, a set of classes, and a set of research groups. There are constraints on how many classes each student takes, and on how many research groups each student belongs. There are also constraints on the maximum number of students per class and on the maximum number of students per group. For each student,

the generator picks random classes to enroll into up to the maximum number of classes per student possible. Similarly, each student is assigned to a random research group.

The nodes in the data graph represent students. There is a `classmates` edge connecting two students for each class they take together, and there is `groupmates` edge if they belong to the same research group. These pieces of information represent observations indicating that two students may be friends, i.e., that they may exhibit a sensitive relationship. The ground truth is generated by computing the probability of a friendship between each two students using the noisy-or model, and assigning the friendship a true value with a probability equal to that likelihood.

The parameters given to the data generator can be varied. We would like to explore graphs which vary in their density, therefore we allow the number of classes and research groups to vary while fixing the number of nodes/students to 100. The constraints on the data are that each student takes two classes, and belongs to one research group. Also, a class can have no more than 25 people, and a group can have no more than 15. We picked probabilities which make sense in the domain. The prior probability of two students knowing each other is $P($`friends`$(v_i, v_j))=0.2$. It is relatively high because the students are from the same department. The probability that two students know each other if they are in the same class c is $P($`friends`(v_i, v_j) —`classmates`$(v_i, v_j, c))=0.4$. The probability that two students know each other if they are in the same research group is $P($`friends`(v_i, v_j) —`groupmates`$(v_i, v_j, c))=0.6$.

8.2 Evaluating privacy preservation in anonymized data

We begin by studying the privacy preservation in the data that results from each of the anonymization techniques. In particular, we study the number of correctly identified sensitive relationships for the following anonymization functions: 1) when the anonymization function leaves the edges between nodes intact (4.2), 2) when it removes 50% of the observations chosen at random (4.2), 3) when it leaves edges between node equivalence classes in the cluster-edge anonymization (4.2), and 4) when it leaves edges between node equivalence classes with a constrained number of observations (4.2). For the last two, each node is assigned randomly to an equivalence class. We vary k, the number of nodes in each equivalence class, and show the results for $k = 2$ and $k = 6$ because they exhibit the tendencies of varying k well.

The data was generated with the default parameters, varying the number of classes and the number of research groups between 10 and 30. A graph, in which there are 10 research groups and 10 classes, is very dense, and a graph at the other extreme with 30 research groups and 30 classes is very sparse. We show these "extreme" cases in Figure 7 and Figure 8. To account for the randomness in the generated graph, we ran the experiments on 100 generated graphs, and present the average performance. Note that when using the default data parameters (at most two classes taken by each student and at most one

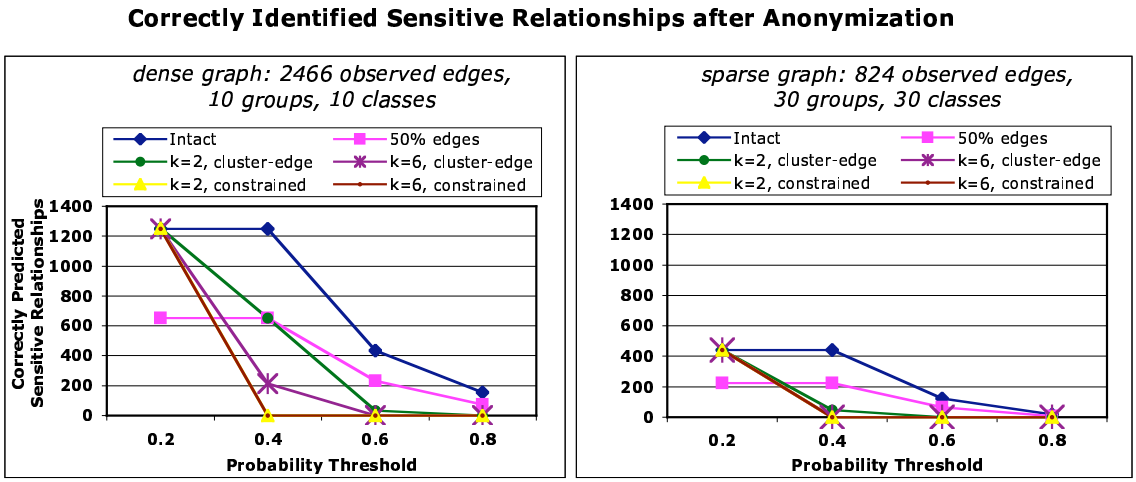

Fig. 7. Comparison between the number of sensitive relationships found after each of six anonymization techniques has been applied. The number of revealed friendships decreases as the friendship likelihood threshold increases. The two constrained cluster-edge methods (at $k = 2$ and $k = 6$) reveal the same number of relationships in both graphs. In the sparse graph, the cluster-edge method at $k = 6$ (not constrained) also overlaps with the two constrained methods.

group of which a student is a member), the maximum possible likelihood for their friendship is 0.89.

We measure the precision, recall rate and the number of inferred sensitive relationships in the anonymized graphs. The precision shows how many of the predicted sensitive relationships are true sensitive relationships. The recall rate measures what portion of the true sensitive relationships can be predicted. Translated into the privacy domain, the recall rate measures what portion of the true sensitive relationships have been compromised, and the precision shows what is the chance that a predicted relationship is really a sensitive one. For example, if the analysis predicts 10 sensitive relationships and only 5 of them are true, then the precision is 0.5. If there are a total of 100 true sensitive relationships in the network, then the recall rate is 0.05. Ideally, a model for predicting sensitive information would should have a high precision and a high recall rate when tested on the original data, and a low precision and a low recall rate when tested on the anonymized data.

A low precision in the anonymized data is more crucial than a low recall rate. A combination of a high precision with a low recall rate in the anonymized data is undesirable because it means that the anonymization can hide most of the sensitive relationships but the ones that can be predicted are highly likely to be true. Results with a low precision and a high recall rate are not as bad. In this

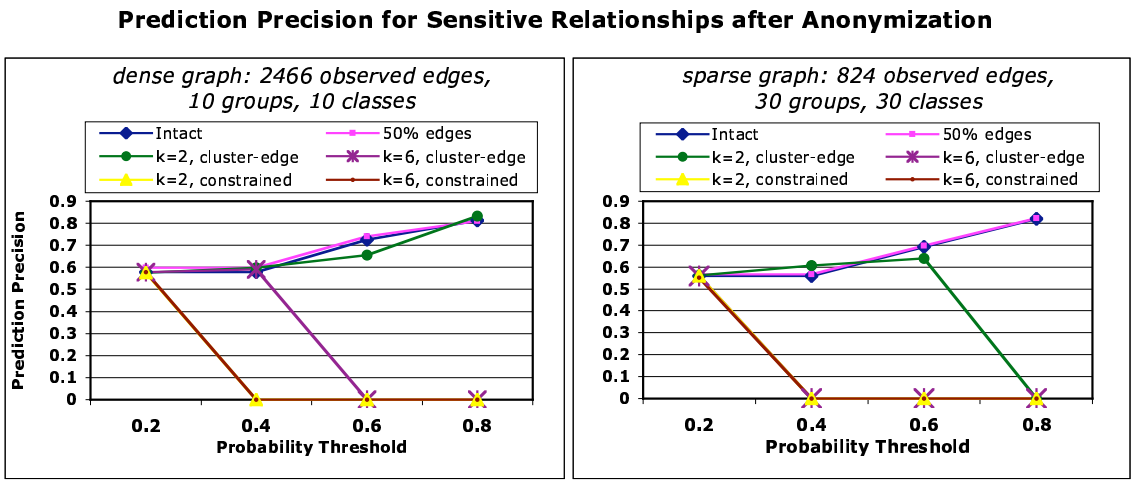

Fig. 8. Comparison between the precision of predicted friendships found after one of six anonymization techniques has been applied. At low threshold values, the number of revealed friendships is large but the precision is low. The precision of the method that removes 50% of the edges at random overlaps with the precision of the intact-edge method in the sparse graph, and nearly overlaps in the dense graph. The precision of the two constrained cluster-edge methods (at $k = 2$ and $k = 6$) overlap as well.

case, even though the anonymization allows many of the true sensitive relationships to be predicted, the true sensitive relationships are indistinguishable from many non-sensitive relationships.

8.3 Results

Figure 7 shows a comparison between the number of sensitive relationships inferred after each of six anonymization techniques has been applied. It shows that at higher thresholds (0.6 and 0.8), keeping all the edges between node equivalence classes preserves privacy much better than deleting 50% of the two-node edges, while having higher utility as discussed in Section 6.3. As expected, for lower k, the privacy preservation is lower: the number of revealed relationships is higher in the data anonymized with the cluster-edge method. In the data anonymized with the cluster-edge method with constraints, varying k yielded to the same results, which is why the graphs of $k = 2$ overlap with the graphs, in which $k = 6$.

We also ran the experiments for other combinations of class and group parameters in the range [10,30]. The experiments confirmed that as the number of observed edges decreases, so does the number of correctly identified sensitive relationships. However, the behavior at different thresholds is proportionately

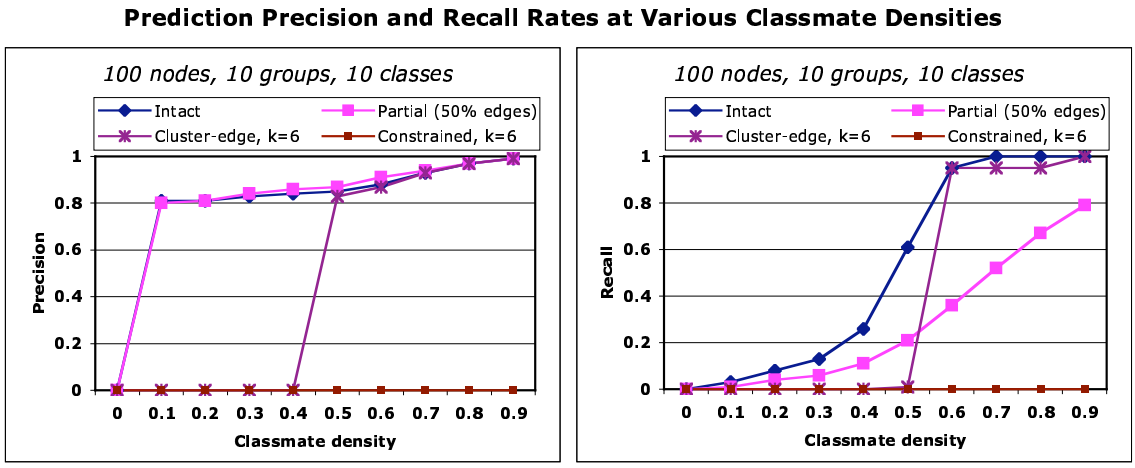

Fig. 9. Comparison between the precision at different classmate density levels (a)) shows that at high density levels, the cluster-edge anonymization preserve privacy as badly as the anonymization which deletes 50 % of the edges. Moreover, the recall rate at these levels (b)) is much higher for the cluster-edge method. The groupmate density is kept constant at 0.1.

the same for all anonymization methods except the cluster-edge method. In the cluster-edge method, the privacy is preserved better in the sparse graph for both k levels, as seen by comparing the dense and the sparse graph results at threshold 0.4. In the sparse graph, the results when $k = 6$ are the same as the ones of the cluster-edge with constraints.

Figure 8 shows that even though lower probability thresholds reveal more sensitive relationships, the precision is low. At higher probability thresholds, the precision is high but on a very small number of predicted relationships.

Experimenting with the number of nodes in the network showed that the precision and sensitivity results were invariant to the network size when the friendship, groupmate and classmate densities were kept constant. The density values were 0.36, 0.1 and 0.2, respectively. The tested networks were of size 100, 200, 300 and 400 nodes. Other constant parameters were the number of groups, 10, the number of classes, 10, and the k-anonymization parameter $k = 6$.

We also varied the multigraph classmate density by varying the number of classes each student joined. Since this parameter was used in the data generator as well, it affected the friendship density of the original graph. The correlation between the two densities was positive. We found that at high classmate density levels the claim that the cluster-edge anonymization preserves privacy better than the anonymization which deletes 50% of the edges no longer held. As Figure 9a) shows that as the class density goes above 0.4 (friendship density is

0.63), the precision of predicted sensitive links is almost the same for the two methods. Moreover, as Figure 9b) at levels above 0.5 (friendship density is 0.76), the data anonymized with the cluster-edge method has much higher recall rate. Again, the number of nodes was 100, the number of groups was 10, the number of classes was 10, and the k-anonymization parameter k was 6.

9 Conclusion

In this paper, we have focused on the problem of link re-identification. We have proposed several approaches for anonymizing graph data and done an initial empirical evaluation of the effectiveness of the different strategies. The work is preliminary, in that we have made very specific assumptions about the model and the data generator parameters. However, because understanding and appreciating the subtleties in the effectiveness of techniques is such an important and timely topic, we hope that this work will motivate further research in the topic.

References

1. G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonimity. *Journal of Privacy Technology*, November 2005.
2. L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x: anonymized social networks, hidden patterns, and structural steganography. In *16th International Conference on World Wide Web (WWW)*, pages 181–190, 2007.
3. R. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *IEEE 21st International Conference on Data Engineering*, April 2005.
4. S. Chawla, C. Dwork, F. Mcsherry, A. Smith, and H. Wee. Toward privacy in public databases. In *Proceedings of the Theory of Cryptography Conference*, 2005.
5. A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *ACM Principles of database systems (PODS)*, pages 211–222, June 2003.
6. L. Getoor and C. P. Diehl. Link mining: a survey. *SIGKDD Explor. Newsl.*, 7(2):3–12, December 2005.
7. M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing social networks, March 2007.
8. N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering*, pages 106–115, April 2007.
9. A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd IEEE International Conference on Data Engineering*, 2006.
10. G. Miklau and D. Suciu. A formal analysis of information disclosure in data exchange. In *ACM Conference on Management of Data (SIGMOD)*, pages 575–586, 2004.
11. M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *26th ACM SIGMOD International Conference on Management of Data*, June 2007.

12. M. E. Nergiz and C. Clifton. Thoughts on k-anonymization. In *IEEE 22nd International Conference on Data Engineering Workshops (ICDEW)*, page 96, April 2006.
13. M. E. Nergiz and C. Clifton. Multirelational k-anonymity. In *IEEE 23rd International Conference on Data Engineering Posters*, April 2007.
14. R. Pang and V. Paxson. A high-level programming environment for packet trace anonymization and transformation. In *ACM SIGSOMM*, 2003.
15. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.
16. P. Samarati. Protecting respondents' identities in microdata release. *Knowledge and Data Engineering*, 13(6):1010–1027, 2001.
17. T. Singliar and M. Hauskrecht. Noisy-or component analysis and its application to link analysis. *Journal of Machine Learning Research*, 7:2189–2213, 2006.
18. L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty*, 10(5):571–588, 2002.