Answering Vertex Aggregate Queries Using Anonymized Social Network Data

Bin Zhou
Department of Information Systems
University of Maryland, Baltimore County
bzhou@umbc.edu

ABSTRACT

As social network data contain rich information about individuals, privacy becomes a critical concern in publishing and exchanging social network data. The existing approaches for privacy-preserving social network data publishing have to modify the local structures of a network substantially which may lead to considerable loss in answering some vertex aggregate queries (e.g., analyzing the degree distribution of vertices). In this paper, we propose a graph partitioning framework to anonymize social network data against degree attacks. A distinct advantage of our approach is that the anonymized social network data can be used to answer some vertex aggregate queries accurately. An empirical study using a large real dataset clearly verifies the effectiveness of our approach.

1. INTRODUCTION

Recently, social network analysis has become more and more important due to the rapid growth of social network data and applications. Many social network analysis tasks involve finding the distribution of some properties (e.g., degree distribution) of a subset of selected vertices. For example, in a disease spreading network, it is interesting to explore the probability that an HIV patient has a neighbor who is also an HIV patient. Such information is very useful in epidemiology research. As another example, it is a popular exercise to analyze the degree distribution of vertices in a large social network. Such queries are called *vertex aggregate queries* (a formal definition in Section 2). Vertex aggregate queries are essential in many social network analysis tasks.

Although social network analysis is highly useful in many applications, privacy is a critical concern in publishing and exchanging social network data. If a social network is published in its raw form, sensitive information may be leaked. With some background knowledge about individuals in a social network such as vertex degrees of target individuals, an adversary may attack the privacy of those victims easily.

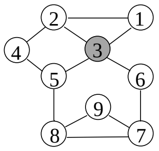

Figure 1: A disease spreading network.

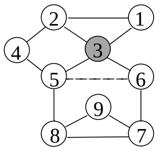

Figure 2: A 2-degree-anonymous network.

Name	ID	Disease
Dell	1	flu
Fred	2	pneumonia
Bob	3	HIV
Harry	4	dyspepsia
Ada	5	HIV
Cathy	6	flu
Ed	7	HIV
Irene	8	pneumonia
George	9	dyspepsia

Table 1: An associated attribute table.

EXAMPLE 1 (DEGREE ATTACKS IN SOCIAL NETWORKS). Consider a synthesized disease spreading network in Figure 1. Each vertex in the network represents one unique individual. Associated with the network, Table 1 records the sensitive label `disease` for each vertex in the network. Two vertices are linked by an edge if the two corresponding individuals have close contact in daily life.

The disease information may be considered as the privacy of the individuals. In order to protect the privacy, when the network data is published, the identifying attributes of each individual including `name` and `SSN` should be removed. Each vertex can be assigned a random number as an ID. Meanwhile, the sensitive information `disease` for each vertex should be retained in the network for the purpose of scientific analysis and research. As a result, the network data to be published is shown in Figure 1, and the associated attribute table is shown in Table 1, where attribute `Name` is only for illustration here and is not for publishing.

Is privacy sufficiently protected in the social network data in Figure 1? As pointed out in [1], a degree attack is often possible. Suppose an attacker knows that Bob appears in the network and has 4 friends also appearing in the network. As vertex 3 is the only one in the network with a degree of 4, the attacker can associate Bob with vertex 3 in the anonymized network in Figure 1 uniquely. Using this association, the attacker can further know that Bob has HIV. □

To protect privacy in social network data, many existing methods [1, 5] conduct anonymization by changing the local structures of vertices in a network in one way or another, such as adding some dummy edges and vertices, randomly

PSOSM'12, April 17 2012, Lyon, France

perturbing the network structures, or generalizing similar edges and vertices into groups. However, these methods are not sufficient in protecting privacy and preserving utility of the social network data.

For example, Liu and Terzi [1] proposed the notion of k-degree anonymity to protect privacy against degree attacks. A graph is said to be *k-degree anonymous* if, for every vertex v, there exist at least $(k-1)$ other vertices in the graph with the same degree as v. Figure 2 is a 2-degree anonymous graph derived from Figure 1. Suppose an attacker knows that Bob is in the network and has 4 friends also in the network. By analyzing the 2-degree anonymous network in Figure 2, the attacker cannot uniquely associate a vertex with Bob since there are two vertices, 3 and 5, with vertex degree 4. However, Table 1 indicates that both vertices have HIV. The attacker still can know that Bob has HIV and thus intrude Bob's privacy. Thus, diversity in anonymization is a critical problem which needs consideration.

In addition, the anonymized networks generated by the existing methods may not be able to answer some vertex aggregate queries accurately. In Figure 1, a simple vertex aggregate query is to ask for the degree distribution of vertices of HIV patients. Figure 2, a 2-degree anonymous version which adds only one edge to Figure 1, cannot provide the exact answer, since an edge connecting vertices 5 and 6 is added, and vertex 5 is an HIV patient.

In this paper, we address the diversity issue in social network anonymization battling degree attacks. Meanwhile, we tackle vertex aggregate queries which are essential in social network analysis.

The rest of the paper is organized as follows. We formalize the problem in Section 2. A graph partitioning framework is illustrated in Section 3. Section 4 presents a systematic empirical study and Section 5 concludes the paper.

2. PROBLEM DEFINITION

We model a social network as a simple graph $G = (V, E, S, \mathcal{S})$, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, S is a set of sensitive labels, and a labeling function $\mathcal{S} : V \to S$ assigns each vertex a sensitive label. For simplicity, we assume that each vertex in a social network carries only one sensitive label. This assumption can be easily removed by expanding the cardinality of the sensitive label set, that is, considering a combination of multiple sensitive labels as a new sensitive label. As analysis on anonymized social network data heavily relies on the sensitive information, we assume that sensitive labels should be retained in anonymization and published in the released data.

In many social network analysis tasks, an essential type of queries is *vertex aggregate query*.

Definition 1 (Vertex aggregate query). In a graph G, the 1-*neighborhood* of a vertex $v \in G_v$, denoted by $Neighbor(v) = \{u | (u, v) \in E(G)\}$, consists of all neighbors of v in G.

A *vertex aggregate query* $Q = (P, f, aggr)$ consists of three components: (1) P is a predicate on the sensitive labels S_G; (2) f is a function on the 1-neighborhood of vertices $Neighbor()$; and (3) aggregate function $aggr$ is optional, if present, $aggr$ aggregates the results of f.

The *answer* to a vertex aggregate query $Q = (P, f, aggr)$ is defined as follows. Let $P_V = \{v \in V(G) | P(v) = true\}$ be the set of vertices in G that satisfy predicate P. If aggregate function $aggr$ is not specified, then the answer to Q is the distribution of $f(Neighbor(v))$ on the set of vertices P_V. If aggregate function $aggr$ is defined, then the answer is $aggr_{v \in P_V}(f(Neighbor(v)))$. □

For example, consider the synthesized disease spreading network in Figure 1. Query $Q_1 = (true, degree())$ finds the degree distribution of vertices in the network, where function $degree(Neighbor(v))$ gives the degree of v (i.e., $|Neighbor(v)|$). In this query, the aggregate function is not specified, and thus the distribution should be returned.

Another query $Q_2 = (HIV(), HIVCnt(), perc())$ finds the percentage of HIV neighbors that an HIV vertex has, where predicate $HIV()$ selects the vertices having label HIV, function $HIVCnt(Neighbor(v)) = \{u \in Neighbor(v) | u \text{ has label HIV}\}$ selects the set of vertices of label HIV in the set $Neighbor(v)$, and $perc()$ is the percentage aggregate function which calculates the ratio of HIV neighbors in the graph, that is, $perc() = \frac{\sum_v |HIVCnt(Neighbor(v))|}{\sum_v |Neighbor(v)|}$, where $v \in HIV()$.

In this paper, we focus on guaranteeing the correctness of using anonymized social network data to answer the above vertex aggregate queries.

As shown in Section 1, diversity of sensitive information is a serious concern in anonymization. We extend the l-diversity principle in relational data [2] to social network data. A social network G is *l-diverse* if there exists an *l-diverse partition* which divides the vertices in V into *equivalence groups* of vertices such that (1) every vertex belongs to one and only one equivalence group; (2) for each equivalence group EG, all vertices in EG have the same degree; and (3) for each equivalence group EG, $\frac{freq(c)}{|EG|} \leq \frac{1}{l}$, where $freq(c)$ is the number of vertices in EG which carry the most frequent sensitive label c in EG, and $|EG|$ is the number of vertices in the equivalence group.

In order to achieve l-diversity in social networks, vertices have to be partitioned into equivalence groups, such that in every equivalence group of vertices, at most $\frac{1}{l}$ of the vertices are associated with the most frequent sensitive label[1]. The larger the value of l, the better the privacy is protected against degree attacks.

3. A GRAPH PARTITIONING FRAMEWORK

Adding edges or deleting edges in anonymization have significant undesirable impact on network structures and social network analysis. Can we avoid adding and deleting any edges but still anonymize a social network to prevent degree attacks from happening?

It has been well recognized in various social networks including Web graph, biological networks and co-author networks that the degrees of vertices in a large social network often follow the power law distribution. According to the power law distribution, the number of vertices with large degrees is often small, which indicates that those vertices are the main targets of degree attacks. To anonymize those vertices into the crowd, it is necessary to modify their degrees in one way or the other.

[1] The original definition of l-diversity in [2] is more general. For simplicity, we adopt the frequency based l-diversity, as most of the previous studies [3, 4] did.

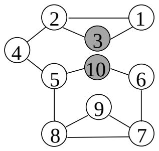

Figure 3: A 2-diverse network.

Name	ID	Disease
Dell	1	flu
Fred	2	pneumonia
Bob	3	HIV
Harry	4	dyspepsia
Ada	5	HIV
Cathy	6	flu
Ed	7	HIV
Irene	8	pneumonia
George	9	dyspepsia
Bob'	10	HIV

Table 2: An attribute table associated with the network.

Group ID	Merging set	# true set
Group1	(3,10)	1
	(1,6)	

Table 3: A matching table associated with the network.

Our idea is to split a vertex of a large degree into several vertices so that each one can be well anonymized. To preserve the local network structure information as much as possible, we should tell users which vertices in the anonymized network are resulted from splitting. Moreover, we should indicate how those split vertices can be "merged" to recover the local structures of the original vertices.

Example 2 (The graph partitioning framework). Consider the social network in Figure 1 again, which does not satisfy 2-diversity.

Vertex 3 is the only one in the network of degree 4. Interestingly, if we split vertex 3 into two, vertex 3 and vertex 10 in Figure 3, the resulted graph (Figure 3) is 2-diverse.

However, splitting vertex 3 in Figure 1 into vertices 3 and 10 in Figure 3 does not preserve the local structure of vertex 3 in the original social network. To preserve the local structure, we can publish a matching table in Table 3. The first row in the table indicates that vertices 3 and 10 can be merged to recover a vertex in the original graph. Vertices 3 and 10 form a merging set.

In order to satisfy 2-diversity, we can add more merging sets into the matching table so that an adversary cannot recover the original graph confidently even if we have to add some noise so that an attacker cannot confidently associate an individual to vertex 3 in Figure 1. By analyzing the social network in Figure 3, we find that vertices 1 and 3, vertices 6 and 10 share similar local network structures. Moreover, vertices 1 and 3, as well as vertices 6 and 10, carry different sensitive labels. We can form an equivalence group using two merging sets, $(3, 10)$ and $(1, 6)$. Merge set $(1, 6)$ acts as the interfering information to help the anonymization.

We can publish a matching table as shown in Table 3.

The table also indicates the number of true sets in each equivalence group. In this group, vertices 3 and 10 should be merged into one with probability $\frac{1}{2}$.

Figure 3, Table 2 and Table 3 can be published as an anonymized social network which is 2-diverse. By analyzing the anonymized network, an attacker with background knowledge of vertex degrees cannot associate a victim with a vertex uniquely, and cannot infer the sensitive information of target victims with probability larger than $\frac{1}{2}$. □

In the anonymization, no edges are added or deleted. The published network can preserve local network structures well. First, the number of vertices in the network is completely preserved, since the number of true sets in each equivalence group is published. Second, all the edges in the network are not changed. There are no dummy edges added into the anonymized network data.

The anonymized network data can be used to answer some aggregate queries with high accuracy. Particularly, it can be used to answer vertex aggregate queries accurately.

Example 3 (Answering vertex aggregate queries). Consider the vertex aggregate queries discussed in Section 2. Query $Q_1 = (true, degree())$ finds the degree distribution of vertices in the network. To answer Q_1 using the released network (Figure 3), the attribute table (Table 2) and the matching table(Table 3), we first generate the degree distribution of the anonymized graph, that is, a degree sequence of $\{2, 3, 2, 2, 3, 2, 3, 3, 2, 2\}$ (from vertex 1 to vertex 10). Next, we scan each group of the merging sets in the matching table to randomly pick up several merging sets. The vertices in each merging sets are merged, and their corresponding degrees are updated in the degree sequence. The number of merging sets to be picked up is exactly equal to the number of true sets in each group. As an example, if the merging set (1,6) is selected, we get a new degree sequence of $\{4, 3, 2, 2, 3, 0, 3, 3, 2, 2\}$ by merging vertex 6 with vertex 1. After removing 0 from the degree sequence, we get the exact degree distribution.

A similar strategy can be used to answer query Q_2 as well. Since query $Q_2 = (HIV(), HIVCnt(), perc())$ finds the percentage of HIV neighbors that an HIV vertex has, we first generate the set of vertices that an HIV vertex has. Thus, we have $HIVCnt(3) = \{1, 2\}$, $HIVCnt(5) = \{4, 8, 10\}$, $HIVCnt(7) = \{6, 8, 9\}$, and $HIVCnt(10) = \{5, 6\}$. We then scan the matching table to randomly pick up the set of merging sets. Again, the number of merging sets to be picked up is exactly equal to the number of true sets in each group. For example, if the merging set (1,6) is selected, we need to update the results in $HIVCnt(3)$, $HIVCnt(7)$, and $HIVCnt(10)$. Assume vertex 6 is merged with vertex 1, we have $HIVCnt(3) = \{1, 2\}$, $HIVCnt(5) = \{4, 8, 10\}$, $HIVCnt(7) = \{1, 8, 9\}$, and $HIVCnt(10) = \{1, 5\}$. So 2 out of 10 have label HIV. Thus, a percentage of 20% is returned, which is exactly the same in the original graph. □

In summary, our method anonymizes a social network by splitting vertices. A matching table is published to facilitate the reconstruction. By using the matching sets in a probabilistic way, a user can recover from the anonymized social network a set of possible social networks. Those possible social networks have the same degree distribution and similar local graph structures as the original social network.

4. EVALUATIONS

We used a real co-authorship dataset from KDD Cup 2003 to examine whether degree attacks with diversity issue may happen in practice. The dataset contains a subset of papers in the high-energy physics section of the arXiv. We extracted author names from the data sources and constructed a co-authorship graph. Each vertex in the graph represents an author, and two vertices are linked by an edge if the two corresponding authors co-authored at least one paper in the dataset. There are $57, 448$ vertices and $120, 640$ edges in the co-authorship graph and the average number of vertex degrees is about 4. To model the sensitive information, we

l	Affiliations as sensitive information
5	4.6%
10	8.2%
15	13.1%
20	19.5%

Table 4: The percentages of vertices violating l-diversity in the co-authorship data.

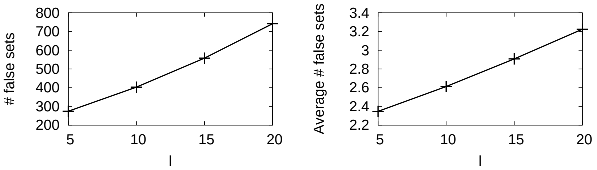

Figure 4: The anonymization cost in anonymized co-authorship data.

extracted the author's affiliation from the raw data. The number of distinct sensitive values is 417.

We placed vertices in the dataset into different chunks, such that each chunk contains vertices with the same vertex degree. A vertex is said to violate l-diversity with respect to degree attacks if an attacker can correctly infer the sensitive label of the vertex with probability larger than $\frac{1}{l}$. Table 4 shows the percentages of vertices violating the l-diversity requirement. The results clearly show that the degree attacks with diversity issue are a real concern for social network data publishing. When the value of l increases, the number of vertices violating the l-diversity requirement increases. This is due to the reason that when l becomes larger, the sensitive labels of vertices in the chunk (with size m) are more likely to appear more than $\frac{m}{l}$ times.

We applied the proposed graph partitioning framework on the co-authorship dataset. We used the total number of false sets and the average number of false sets to measure the anonymization cost. The results are reported in Figure 4. In general, the cost is quite reasonable.

We also examined the query answering performance using the anonymized data. As discussed in Section 3, vertex aggregate queries can be answered using the anonymized data generated by our method exactly. All the other existing approaches cannot achieve such a strong result.

In addition, to examine the efficiency of our anonymization method, we measured several graph properties using the original graph data and the anonymized data. In the anonymized data, the false sets in the matching table introduce uncertainty into graph structures. All the merging sets in an equivalence group are considered to have the same probability to be merged. To re-construct the original graph, we randomly selected a set of merging sets with size equaling to the number of true sets in that equivalence group. We considered several graph property measures, including the clustering co-efficient (CC) and the average path length (APL). Figure 5 shows the results in the original graph and the anonymized graph using the proposed partitioning framework. The clustering co-efficient and the average path length decrease while l increases, this is because when vertices are merged, the graph shrinks.

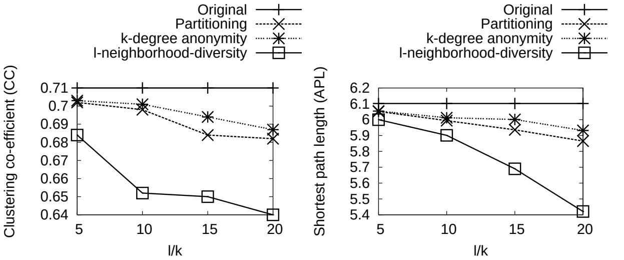

Figure 5: The data utility in anonymized co-authorship data.

To make a comparison between our method and some previous studies, we considered the k-degree anonymous graph in [1] and the l-diversity neighborhood graph in [5]. In [1], the graph is constructed by adding/deleting edges to achieve a k-anonymous degree sequence so as to prevent degree attacks, while in [5], the graph is generated by greedily adding edges so as to achieve l-diversity, that is, for each equivalence group in which vertices have isomorphic 1-neighborhood subgraphs, the most frequent sensitive label appears at most $\frac{m}{l}$ times, where m is the size of the equivalence group. We implemented the two methods, then generated the anonymized graphs separately, and calculated the clustering co-efficient and the average path length. The results are shown in Figure 5. The k-degree-anonymous graph has the most similar clustering co-efficient and the average path length. The graph generated using partitioning has a comparable results. Moreover, the graph generated using partitioning considers l-diversity, which is a stronger privacy requirement than k-degree anonymity. The graph generated using the method in [5] has the worst result, which may be due to the reason that the background knowledge considered in [5] is 1-neighborhood structures, which is much stronger than the background knowledge of vertex degrees. Thus the anonymized graph has to lose more utility to achieve privacy.

5. CONCLUSIONS

In this paper, we addressed the diversity issue in social network anonymization. We proposed a graph partitioning framework to anonymize the network, which can retain all the direct relationships of vertices in the network. The experimental results indicated that the anonymized network generated using our methods can answer some vertex aggregate queries accurately.

There are several interesting directions to explore in the future. For example, in this paper we modeled the complete vertex degree as the background knowledge. In practice, partial background knowledge of the vertex degrees is more realistic. This introduces new challenges for background knowledge modeling, meanwhile opens new opportunities to improve the utility of the anonymized social network data.

6. REFERENCES

[1] K. Liu and E. Terzi. Towards identity anonymization on graphs. In SIGMOD'08, pages 93–106, New York, NY, USA, 2008. ACM Press.

[2] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. In ICDE'06, Washington, DC, USA, 2006. IEEE Computer Society.

[3] X. Xiao and Y. Tao. Anatomy: Simple and effective privacy preservation. In VLDB'06, pages 139–150. ACM, 2006.

[4] X. Xiao and Y. Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In SIGMOD'07, pages 689–700, New York, NY, USA, 2007. ACM.

[5] B. Zhou and J. Pei. The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowledge and Information Systems*, 28(1):47–77, 2011.