Noname manuscript No.
(will be inserted by the editor)

Neighborhood Randomization for Link Privacy in Social Network Analysis

Amin Milani Fard · Ke Wang

Received: date / Accepted: date

Abstract Social network analysis has many important applications but it depends on sharing and publishing the underlying graph. Link privacy requires limiting the ability of an adversary to infer the presence of a sensitive link between two individuals in the published social network graph. A standard technique for achieving link privacy is to probabilistically randomize a link over the space for node pairs. A major drawback of such graph-wise randomization is that it ignores the structural proximity of nodes, thus, alters considerably the structure of social networks and distorts the accuracy of social network analysis. To address this problem, we propose a structure-aware randomization scheme, called *neighborhood randomization.* This scheme models a social network as a *directed* graph and probabilistically randomizes the *destination* of a link within a local *neighborhood.* By confining the randomization to a local neighborhood, this scheme drastically reduces the distortion to the graph structure yet hides a sensitive link. The trade-off between privacy and utility is dictated by the retention probability of a destination and by the size of the randomization neighborhood. We conduct extensive experiments to evaluate this trade-off using real life social network data.

Keywords Social network analysis · Privacy-preserving data publishing · Link perturbation

A. Milani Fard
School of Computing Science, Simon Fraser University, Canada
Now at the University of British Columbia, Canada
E-mail: aminmf@ece.ubc.ca

K. Wang
School of Computing Science, Simon Fraser University, Canada
E-mail: wangk@cs.sfu.ca

1 Introduction

Social network analysis is gaining attentions and applications in database and data mining, web applications, sociology, and communication. Social networks vary from social networking sites such as Facebook, LinkedIn, and Twitter, to user interaction networks such as emails, chats, blogs, and file sharing system [22]. The first step in social network analysis is sharing the network data with the data miner, however, this raises privacy concerns due to disclosure of sensitive relationships among social networks users. Privacy preserving social network analysis goes beyond removing identifying attributes of users from their nodes because an adversary could apply background knowledge about a local neighborhood of a node such as degree to re-identify the node of a user [2]. Another challenge is preserving graph utility. Simple statistics on local features such as the number of nodes/links do not capture structural properties such as centrality, shortest distances, and communities that are essential for social network analysis [24]. Preserving user privacy while retaining vital structural properties is the focus of this work.

Several types of privacy disclosure for social network data have been studied in the literature, namely, content disclosure, identity disclosure, and link disclosure. *Content disclosure* refers to disclosure of content information associated with each user, such as age, gender, sex orientation, and are typically addressed by data anonymization (see works in [11]). Even if no content information is released, *identity disclosure* can occur where the node of a target user is re-identified using knowledge on neighborhood structure such as the degree of a node [2]. Identity disclosure can be limited by node anonymization, which renders a neighborhood of several nodes similar, such as degree k-anonymity [14,19,29,30], k-automorphism [31], and k-isomorphism [8]. Often, achieving neighborhood similarity requires considerable structural distortion, such as transforming a graph into k disconnected isomorphic subgraphs required by the k-isomorphism technique. *Link disclosure* occurs when an adversary infers the existence of a link between two users with a high probability. The study in [8] shows that limiting identity disclosure is insufficient for limiting link disclosure. Link disclosure is the focus of this work.

1.1 Motivation

In some scenarios the contents or identities of nodes in a network are not sensitive whereas the links between nodes are considered to be confidential. Examples are financial transaction networks, email networks, and professional social networks in which existence of links, i.e., money transactions, private emails, or friendship between two individuals and their direction are considered sensitive.

In this paper, we consider the problem of publishing a sanitized version of a social network graph for data analysis while limiting link disclosure. Specifically, we want to ensure that an adversary cannot reliably infer the presence of a true link in the original graph given the presence of the link in the sanitized graph. We do not consider inferring the absence of a link because in a typically sparse social network graph the absence of a link between a pair of nodes is far more common than the presence of a link of the pair. In this case, privacy concerns are primarily caused by the presence of a link. For a similar reason, the absence of a link in the sanitized graph most likely implies the absence of the link in the original graph, so we do not consider inferring the presence of a true link from the absence of a link in the sanitized graph.

A standard technique for preserving link privacy is *link randomization* [13, 26]. This technique, called *graph-wise randomization* in this paper, models a social network as an undirected graph, and randomly deletes some fraction of existing links (u, v) and adds the same number of non-existing links (w, z) randomly chosen from the space of all non-existing edges. This operation is equivalent to randomizing a true link (u, v) to a false link (w, z).

The major drawback of graph-wise randomization approach is considerable structural distortion because the false link (w, z) is chosen at random from the entire space for non-existing links. For example, adding a false link between two remote nodes, or deleting the only link connecting two parts of the graph, drastically affects the shortest path and reachability analysis. In addition, this approach treats a social network as an undirected graph, whereas many real-world social networks, such as Twitter followers, e-mail networks, Google+ circles, and Facebook, are inherently directed graphs. For example, an edge (u, v) may represent that "u endorses v", "u cites v", and "u follows v", which obviously differs from "v endorses u", "v cites u", and "v follows u" in terms of both privacy and utility. Indeed, it has a different privacy and utility implication to publish a user as the source of a link or as the destination of a link. In the case of directed graphs, ignoring the direction of links would distort the graph to the extent that the resulting graph is useless. If data analysis is required to be based on directed graphs, there is no choice but to consider randomizing directed graphs. To that end it suffices to hide either the source or the destination node of a link.

1.2 Contributions

This paper makes the following contributions.

Contribution 1: sanitation operator. We propose a structure-aware randomization scheme, called *neighborhood randomization*, with two distinctive features. First, it models a social network as a *directed* graph, and for each link (u, v), probabilistically retains the *destination* v with a certain probability p and replaces the destination with a random node w with probability $1 - p$. Second, it picks the randomized destination w from a local *neighborhood* of u. This scheme better preserves the graph structure by keeping the source u of a link intact and confining the random destination w to a local neighborhood of the source u.

Contribution 2: link privacy. We formalize the notion of link privacy by ensuring the probability that an observed link in the sanitized graph is a true link is no more than $1 - \delta$. The publisher-specified parameter δ represents a trade-off between the uncertainty of links, which is required for link privacy, and the preservation of graph structure, which is required by graph utility.

Contribution 3: algorithm. We present a sanitation algorithm based on neighborhood randomization. For a given level δ of link privacy, this algorithm preserves graph structures by maximizing the link retention probability p and minimizing the size of randomization neighborhood. We analyze the defense of this approach against an adversary with additional background knowledge, and evaluate the effectiveness of preserving vital metrics for social network analysis through extensive experiments on real life social network data.

The rest of the paper is organized as follows. We review related work in Section 2, define the problem in Section 3, present the neighborhood randomization approach in Section 4, and evaluate the utility of sanitized graphs in Section 5. Finally, we conclude the paper.

2 Related Work

Most previous works consider identity privacy, e.g., [3,19,30,31]. The study in [8] showed that limiting identity disclosure is insufficient for limiting link disclosure. We focus on the work on limiting link disclosure, which can be divided into three major groups.

The first group achieves some form of *edge anonymity* by transforming the graph to have some structural similarity. In particular, k-isomorphism [8] transforms the original graph into k disconnected pairwise isomorphic subgraphs through link insertion and deletion, and [28] partitions nodes into equivalence classes and inducing edge equivalence classes between node classes. In general, considerable structural distortion is required to provide the required structural similarity.

The second group of work performs some form of *structure collapsing* in order to hide sensitive links. In [7,14,29], the graph is partitioned into clusters and each cluster is collapsed into one super-node. Although these methods store statistic information about the nodes in super-nodes and the number of edges between clusters, the structure among the nodes represented by a super-node is lost due to the cluster collapsing.

The third group is based on *link perturbation.* Our work belongs to this group. The work in [26,13] randomly adds m non-existing links and randomly deletes m existing links, or randomly switches m pairs of links. As explained in Section 1.1, such link perturbation introduces considerable structural distortion due to its insensitivity to structural proximity. The work in [26] presented a randomization method to preserve spectral characteristics of graphs, but there is a lack of formal privacy measures. To address structural distortion, we proposed *subgraph-wise perturbation* [21] that partitions the graph into subgraphs and randomizes the links within each subgraph. However, the graph partitioning introduces new threats in that a node may become a much more popular destination in a subgraph than in the original graph, which increases the risk of being inferred as the destination of a link in the subgraph. As a solution to this issue, a degree balancing phase is applied which moves edges between subgraphs, thus, compromises the effectiveness of confining the randomization to each subgraph. Our new approach, *neighborhood randomization*, does not partition the graph, therefore, does not have this problem.

The work in [2] describes a family of *subgraph attacks* in which an adversary learns the existence of a link between a targeted pair of nodes by constructing a distinguishable subgraph with edges attached to the targeted nodes. They present both *active* and *passive* attacks on anonymized social networks, showing that both types of attacks can be used to reveal the identity of a targeted node. The work in [17] studies *link prediction based attacks* that exploit certain local graph features to infer the existence of a sensitive link. The aforementioned attacks heavily depend on identifying certain subgraphs or collecting graph features. Our neighborhood randomization deters such attacks because subgraphs or local features cannot be reliably identified or collected from randomized links. Besides, we do not assume that an adversary never identifies a target individual; rather, our privacy goal is to hide the existence of a link between two individuals by bounding the probability of inferring the true destination of an observed link. Compared to deterministic algorithms, randomization-based algorithms are less vulnerable to the attack that exploits the knowledge on the algorithms, such as *minimality attack* [25].

In [20] the risks of sequential releases of the same social network is proposed. They introduce the *degree-trail attack*, which compares the degrees of the nodes in the published graphs with the degree evolution of the target node for re-identification. As a solution they propose a variation of link perturbation [13], called stable link randomization, which reuses the randomized edges from prior publications and performs randomization only on new edges/non-edges. In this work we do not consider the sequential release problem.

3 Problem Statement

This section defines the data model for social networks, privacy notion, and the problems studied. In this work, we consider a social network that can be represented by a *simple directed* graph $G = (V, E)$, where V is the set of nodes $\{1,\dots,|V|\}$ representing the network users, and E is the link table with two columns (Src, Dst). A link (u, v) in E represents a directed relationship from user u to user v. u is the *source* of the link and v is the *destination* of the link. If a social network has an undirected edge (such as the "friend-of" relationship), we can regard any of the two nodes in an edge as the source and other node as the destination. $Dst(u)$ denotes the set of destinations of a source u. $Dst(G)$ denotes the set of all destinations and $Src(G)$ denotes the set of all sources in G. There is a *path* $u_0, u_1, \cdots, u_q$ from u_0 to u_q, or a node u_q is *reachable* from a node u_0, if for $0 \leq i \leq q-1$, (u_i, u_{i+1}) is a link in G. q

is the *length* of the path. The *distance* from u_0 to u_q is the length of a shortest path from u_0 to u_q.

We consider the *data publishing* problem where a publisher wants to release a sanitized version of G, denoted by G^*, to serve a variety of data analysis, ranging from *PageRank* [6] to common graph metrics such as *degree centrality, closeness centrality, betweenness centrality, transitivity, eigenvalues, average shortest path length*, etc. [24]. Such metrics are important for understanding the trends in relationships, communities, information spreading, influential users, etc. We consider it a privacy breach to infer the existence of a link between two users. We assume that all content information about a node, such as the user's name, age, gender, have been removed from G^*. We protect user's privacy by limiting the ability of an adversary to infer the existence of a link between two nodes in G, given the published graph G^*.

More precisely, suppose that G^* is obtained from G by probabilistically randomizing each link in G to a false (i.e., non-existing in G) link following a fixed probability distribution p. The detail of this randomization operator will be discussed in Section 4. Let $A = (a_{ij})_{|V|\times|V|}$ and $A^* = (a^*_{ij})_{|V|\times|V|}$ be the adjacency matrix of G and G^* respectively. $a_{ij} = 1$ means that (i, j) is a link in G, and $a^*_{ij} = 1$ means that (i, j) is a link in G^*. On observing a link (i, j) in G^*, the adversary tries to infer if (i, j) is a link in G. An observed link (i, j) in G^* has two possibilities: (i, j) is a link in G if it was retained by the randomization operator, and (i, j) is not a link in G if it was added by the randomization operator. Let $Pr[a_{ij}{=}1|a^*_{ij}{=}1]$ denote the probability that (i, j) is a link in G given that (i, j) is observed in G^*. To limit the adversary's ability to infer the presence of a link in G, we want to bound $Pr[a_{ij}{=}1|a^*_{ij}{=}1]$. With each link in G being retained with the probability p and being replaced with a false link with the probability $1-p$, the best guess for $Pr[a_{ij}{=}1|a^*_{ij}{=}1]$ is the percentage of true links (i.e., retained links) among all observed links in G^*. This percentage is p. So $Pr[a_{ij} = 1|a^*_{ij} = 1] = p$.

Definition 1 (Perturbation-Privacy) We say G^* is *δ-perturbation-private* if $Pr[a_{ij} = 1|a^*_{ij} = 1] \leq 1 - \delta$ for $1 \leq i \neq j \leq |V|$, and $0 \leq \delta \leq 1$. □

Several points are worth mentioning. First, as explained in Section 1.1, we do not consider inferring the absence of a link in G because privacy concern primarily comes from the presence of a link, which is because the absence of a link is far more common than the presence of a link in a typically sparse social network. For a similar reason, we do not consider inferring the presence of a link in G from the absence of a link in G^* because the absence of a link between a pair of nodes in G^* most likely originates from the absence of the link in G. Second, the above privacy definition does not consider the adversary's background knowledge on a neighborhood structure such as the degree of a node. One reason is that such background knowledge relies on identifying matching structures, but our randomization operator makes this identification less reliable by introducing enough uncertainty to each link. We will discuss this in details in Section 4.3.2 after introducing our randomization operator.

Definition 2 (Link Perturbation Problem) Given a graph G and $0 \leq \delta \leq 1$, we want to produce a sanitized graph G^* that is δ-perturbation-private while preserving, as much as possible, utility for social network analysis. □

4 Neighborhood Randomization

We present our solution to the link perturbation problem. We first present an overview, then the choice of neighborhood for randomization, a key issue in our approach, and finally the algorithm and analysis.

4.1 Overview of the Method

The traditional link perturbation [26,13] randomly replaces an existing link with a randomly selected non-existing link. Since the replacing link is selected randomly, it ca be "structurally remote" from the replaced link, thus, the resulting graph tends to be a random graph. We address this issue with three ideas.

First, to hide a link it suffices to hide either its destination or its source because knowing one but not the other does not help to infer the existence of the link. We consider hiding the destination, but the same method can be used to hide the source. Second, to hide a destination we can retain the destination v of a link (u, v) with some probability p and replace it with a false destination w with probability $1 - p$. The retention probability p depends on the desired δ-perturbation-privacy. Third, the false destination w should be chosen from a local neighborhood of the source u so that it is structurally close from u to preserve the graph structure.

These ideas are incorporated in the *neighborhood randomization* defined below. For now, let $DDS(u)$ denote a set of candidate nodes for the randomized destination w in some neighborhood of u, called the *Destination Decoy Set* for u. We will formally define $DDS(u)$ in Section 4.2.

Neighborhood Randomization. Given the retention probability p and the destination decoy set $DDS(u)$ for each source u in G, for each link (u, v) in G, we toss a coin with head probability p. If the coin lands on head, we retain the link (u, v), otherwise, we replace the link with a false link (u, w), where w is randomly chosen from $DDS(u)$ without replacement. Note that the decision for each link G is made independently and that the outcome of each coin toss (i.e., head or tail) is hidden from the adversary.

The choice of p and $DDS(u)$ dictates the trade-off between privacy and utility. The larger the p is, the more true links are retained, but at the same time, the more likely the adversary can infer a true destination of a link. The more compact the neighborhood for defining $DDS(u)$ is, the closer the randomized destination is to the source u and the more graph structure is preserved, but at the same time, the adversary can infer that an observed destination is a close neighbor of the source. In the rest of this section, we discuss the choices for $DDS(u)$ and p in detail.

4.2 Choosing Destination Decoy Sets

The choice of $DDS(u)$ is dictated by the following requirements.

Requirement 1. (Structural proximity) The randomized destination w of a link (u, v) should be structurally close to the source u so that a false link does not connect two structurally remote nodes.

Requirement 2. (False link) The randomized link (u, w) should not be a true link in G. To satisfy this requirement, $DDS(u)$ should not contain any node from $Dst(u)$.

Requirement 3. (Indistinguishability) The randomized link (u, w) should not be a self-loop or multi-link. This is because a self-loop or a multi-link in G^* can be immediately distinguished from a true link since G does not contain any self-loop or multi-link. To avoid a self-loop, u should be excluded from $DDS(u)$, and to avoid multi-link originating at u, all randomized destinations w of the same source u must be distinct.

We say that a randomized link (u, w) satisfying Requirement 1 is *(structurally) close* and a randomized link (u, w) satisfying Requirements 2 and 3 is *(structurally) legitimate*. To address these requirements, we introduce the notion of r-neighborhood.

Definition 3 (r-neighborhood) For an integer $r \geq 0$ and a source u in G, the *r-neighborhood* of u, denoted by $N_r(u)$, contains all destinations in G that are reachable from u within the distance (shortest path length of) r. r is called the *radius*. $N_*(u)$ contains all destinations that are reachable from u by any finite distance. □

Note $N_0(u) = \{u\}$, $N_1(u) = \{u\} \cup Dst(u)$, and for $r \geq 1$, $N_r(u) = \{u\} \cup (\cup_{v \in Dst(u)} N_{k-1}(v))$. Requirement 1 implies that the nodes in $DDS(u)$ should be in $N_r(u)$ for some "small" r. Requirements 2 and 3 imply that $DDS(u)$ should not contain any node in $N_1(u)$ and a randomized destination w for u should be sampled from $DDS(u)$ *without replacement*. Thus the nodes in $DDS(u)$ are selected from $N_r(u) - N_1(u)$, where $r > 1$. Since u has $|Dst(u)|$ out-going links and since there is a non-zero probability (assuming $1 - p > 0$) that each link will be replaced with a randomized link, sampling without replacement entails that $DDS(u)$ must contain at least distinct $|Dst(u)|$ nodes, i.e, $|Dst(u)| \leq |DDS(u)|$.

$DDS(u)$ satisfying the above requirements can be specified by $DDS(u, r, s)$, where r specifies the "radius" of the neighborhood for $DDS(u)$ and s specifies the "size" of $DDS(u)$. From the above discussion, $r > 1$ and $|Dst(u)| \leq s$.

DDS(u,r,s). The following procedure chooses s nodes for $DDS(u, r, s)$ according to a priority corresponding to the closeness to u. Let $s_1 = |N_r(u)| - |N_1(u)|$, $s_2 = |N_*(u)| - |N_1(u)|$, and $s_3 = |Dst(G)| - |N_1(u)|$. Note $s_3 \geq s_2 \geq s_1$.

Case 1. If $s_1 \geq s$, there are enough number of nodes in $N_r(u) - N_1(u)$, so $DDS(u, r, s)$ contains s nodes randomly selected from $N_r(u) - N_1(u)$.

Src	Dst
1	4
2	1
2	3
3	6
4	2
4	5
5	4
5	6
5	7

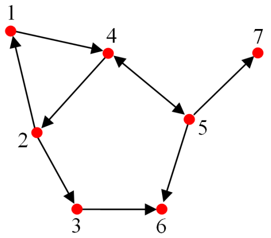


Fig. 1 A simple social network graph

Table 1 r-neighborhood of sources in Figure 1

u	$N_1(u)$	$N_2(u)$	$N_3(u)$	$N_4(u)$
1	{1,4}	{1,4,2,5}	{1,4,2,5,3,6,7}	$N_3(1)$
2	{2,1,3}	{2,1,3,4,6}	{2,1,3,4,6,5}	{2,1,3,4,6,5,7}
3	{3,6}	$N_1(3)$	$N_1(3)$	$N_1(3)$

Case 2. Else if $s_2 \geq s$, there are not enough number of nodes in $N_r(u) - N_1(u)$ but there are enough number of destinations reachable from u. In this case, $DDS(u, r, s)$ contains all nodes from $N_r(u) - N_1(u)$ and contains $s - s_1$ nodes randomly selected from $N_r(u) - N_r(u)$, where r is the smallest integer $\geq r$ such that $|N_r(u)| - |N_r(u)| \geq s - s_1$.

Case 3. Else if $|Dst(G)| - |N_*(u)| \geq s - s_2$, there are not enough number of reachable destinations from u but there are enough number of nodes if non-reachable destinations are used. In this case, $DDS(u, r, s)$ contains all (reachable) nodes from $N_*(u) - N_1(u)$ and contains $s - s_2$ nodes randomly selected from $Dst(G) - N_*(u)$.

Case 4. Else if $|V| - |Dst(G)| \geq s - s_3$, there are not enough number of destinations but there are enough number of nodes if non-destinations are used. In this case, $DDS(u, r, s)$ contains all nodes from $Dst(G) - N_1(u)$ and contains $s - s_3$ nodes randomly chosen from $V -$ $Dst(G)$.

Note that if $s \leq |V| - |N_1(u)|$, we have $|V| - |Dst(G)| \geq s - s_3$, so the above covers all cases and $DDS(u, r, s)$ is well defined.

Example 1 Table 1 shows the r-neighborhoods of three source nodes $\{1, 2, 3\}$ of the graph depicted in Figure 1. $Dst(G) = \{1, 2, 3, 4, 5, 6, 7\}$. Let $r = 2$ and $s = 2$. Consider the source $u = 1$. $s_1 = |N_2(u)| - |N_1(u)| = 2$, and $s_1 \geq s$. This is Case 1. So $DDS(u, r, s)$ contains two nodes from $N_2(u) - N_1(u)$, i.e., $\{2, 5\}$.

Consider the source $u = 3$ (for $r = 2$ and $s = 2$). $s_1 = |N_2(u)| - |N_1(u)| = 0$, $s_1 < s$. $s_2 = |N_*(u)| - |N_1(u)| = 0$, so $s_2 < s$. $|Dst(G)| - |N_*(u)| = 5 \geq s - s_2$. This is Case 3. So $DDS(u, r, s)$ contains all nodes in $N_*(u) - N_1(u)$, which is empty, and two nodes from $Dst(G) - N_*(u) = \{1, 2, 4, 5, 7\}$. □

$DDS(u, r, s)$ will not be published. From the published G^*, the adversary knows that each destination of u in G^* is either in $Dst(u)$ or in $DDS(u, r, s)$, but does not know which with certainty. For this reason, hiding $Dst(u)$ requires hiding $DDS(u, r, s)$ for each source u.

Consider Figure 1 as part of a financial transaction network. Assume that in the published graph link (1,4) is perturbed into (1,5). If an attacker can identify Bob is node 1, Alice is node 4, and Tom is node 5, he can not infer the true transaction from Bob to Alice or detect the fake transaction from Bob to Tom with high certainty.

A smaller s and r means that a randomized destination is chosen from a more compact neighborhood and with fewer choices, thus, better preservation of graph structure. However, a larger s and r means more uncertainty for a randomized destination, thus better hiding the true destination. The data publisher can use these parameters to balance between structural preservation and link privacy.

The next theorem gives a condition for guaranteeing that every randomized link (u, w) is legitimate, i.e., satisfying Requirements 2 and 3.

Theorem 1 *Let $r \geq 2$ and $p > 0$, where p is the retention probability of a destination. Every randomized link (u, w), where w is sampled from $DDS(u, r, s)$ without replacement, is legitimate if and only if $s \geq |Dst(u)|$.*

Proof (If) Since $DDS(u, r, s)$ does not contain any node in $N_1(u)$, for any w in $DDS(u, r, s)$, (u, w) is not a true link or a self-loop. u has at most $|Dst(u)|$ randomized links (u, w), where w is sampled from $DDS(u, r, s)$ without replacement. Since $DDS(u, r, s)$ contains $s \geq |Dst(u)|$ nodes, sampling without replacement ensures that the destination w of every randomized link (u, w) is distinct, thus, (u, w) is not a multi-link. This shows that (u, w) is legitimate.

(Only if) For a non-zero retention probability p, there is a non-zero probability that every link (u, v) in G from the source u is replaced with a randomized link (u, w). In this case, every randomized link (u, w) is legitimate only if $s \geq |Dst(u)|$ holds because all w must be distinct. □

The next corollary summarizes when $DDS(u, r, s)$ is defined and every randomized link is legitimate.

Corollary 1 *Assume $|V| \geq 2|Dst(u)| + 1$. For any $r \geq 2$ and $|Dst(u)| \leq s \leq |V| - |Dst(u)| - 1$, $DDS(u, r, s)$ is defined and every randomized link (u, w) is legitimate, where w is sampled from $DDS(u, r, s)$ without replacement.*

Proof $s \leq |V| - |Dst(u)| - 1$ implies $|V| \geq s + |Dst(u)| + 1 = s + |N_1(u)|$, which implies the condition $|V| - |Dst(G)| \geq s - s_3$ in Case 4 of $DDS(u, km)$. Therefore, $DDS(u, r, s)$ is defined. The rest of the proof follows from Theorem 1 and $|Dst(u)| \leq s$. □

Let us consider a case where the condition in Corollary 1 fails. Suppose that a source u has every other node as its destination in G, i.e., $|Dst(u)| = |V| - 1$, so the condition $|V| \geq 2|Dst(u)| + 1$ in Corollary 1 fails. In this case, $N_1(u) = V$, and since $DDS(u, r, s)$ must not contain any nodes in $N_1(u)$ (i.e., Requirements 2 and 3), $DDS(u, r, s)$ is undefined for $s > 1$ (i.e., none of the four cases holds). In this case, randomizing a link (u, v) to any link (u, w) will create a self-loop (if $w = u$) or a multi-link (if $w \neq u$) in G^*. This situation happens because u has too many destinations. To prevent this situation, the condition $|V| \geq 2|Dst(u)|+1$ states that a source u is connected to no more than half of all nodes in G. This ensures that for s such that $|Dst(u)| \leq s \leq |V| - |Dst(u)| - 1$, $DDS(u, r, s)$ is defined. For a large social network graph, it is almost certain that $|V| \geq 2|Dst(u)| + 1$ holds due to the well known link sparsity for social networks. For this reason, we assume that $DDS(u, r, s)$ is defined. Indeed, for all real life social networks considered in Section 5, $|V| \geq 2|Dst(u)| + 1$ always holds.

4.3 Algorithm and Privacy Analysis

In this section, we present the algorithm for neighborhood randomization and analyze the privacy implication.

4.3.1 Algorithm

Algorithm 1 presents the neighborhood randomization algorithm. The input consists of the graph G, the link retention probability p, the neighborhood radius r, and the size $s(u)$ for $DDS(u, r, s)$ for each source node u. We assume that the condition in Corollary 1 holds. The algorithm produces the sanitized graph G^* as follows. For each source u, it computes $DDS(u, r, s(u))$ and randomizes the destination of every link (u, v) with the retention probability p (Lines 4-11). In particular, for each link (u, v), it tosses a coin with head probability p, adds the link (u, v) to G^* if the coin lands on head, or adds a new link (u, w) to G^* if the coin lands on tail, where w is sampled from $DDS(u, r, s(u))$ at random without replacement. After considering all sources u in G, the algorithm returns G^*. Note that $DDS(u, r, s(u))$ is computed with respect to the input graph G, which is unaffected by the randomization of a link, so the order of considering the sources u at Line 2 does not affect $DDS(u, r, s(u))$.

The time complexity consists of two parts. The first part comes from randomizing each edge in G, which takes $O(|E|)$ time. The second part comes from computing $DDS(u, r, s(u)$ for each source node u (i.e., Line 3), which is dominated by the time of computing the r-neighborhood of u, $N_r(u)$. Since r is typically small due to the structural proximity requirement (i.e., Requirement 1), say 2-3, this part can also be computed efficiently. For the rest of this section, we focus on privacy analysis.

4.3.2 Privacy Analysis

Let us analyze what an adversary can learn from G^* produced by Algorithm 1. We assume that Algorithm 1 and the parameters r, s, p are public and thus known to the attacker. $DDS(u, r, s)$ will not be published. The next theorem shows that G^* is δ-perturbation-private if the retention probability

Algorithm. 1 Neighborhood Randomization

Input: A directed graph G, link retention probability p, neighborhood radius $r \geq 2$, the size $s(u)$ of $DDS(u, r, s)$ for all sources u. Assume the condition in Corollary 1 holds.
Output: The sanitized graph G^*

```
$G^* \leftarrow \varnothing$
for each source node $u$ in $G$ do
    compute $DDS(u, r, s(u))$
    for each link $(u, v) \in G$ do
        toss the coin with head probability $p$
        if the coin lands on head then
            add $(u, v)$ to $G^*$
        else
            add $(u, w)$ to $G^*$ with $w$ randomly sampled from $DDS(u, r, s(u))$ without replacement
        end if
    end for
end for
return $G^*$
```

p is set to $1 - \delta$ and this is the maximum retention probability for providing δ-perturbation-privacy.

Theorem 2 *Assume $|Dst(G)| \geq 2|Dst(u)| + 1$ for every source u. Let $r{\geq}2$ and $|Dst(u)|{\leq}s(u){\leq}|Dst(G)|{-}|Dst(u)|{-}1$. Let G^* be produced by Algorithm 1 with $p = 1 - \delta$. Then (i) every randomized link (u, w) in G^* is legitimate, (ii) G^* is δ-perturbation-private, and (iii) $p = 1 - \delta$ is the maximum retention probability for ensuring δ-perturbation-privacy.*

Proof (i). This follows from Corollary 1. (ii). From Definition 1, G^* is δ-perturbation-private if the fraction of true links in G^* is at most $1{-}\delta$. With the retention probability $p = 1 - \delta$, each link (u, v) in G is retained with probability $1{-}\delta$ and is randomized to a link (u, w) with probability δ. Since every randomized (u, w) is legitimate, thus, a false link, the fraction of true links remaining in G^* is $1{-}\delta$, so G^* is δ-perturbation-private. (iii). Any retention probability $p > 1{-}\delta$ will lead to a larger fraction of true links in G^*. Thus $1{-}\delta$ is the maximum retention probability for ensuring δ-perturbation-privacy. □

Other than learning the information permitted by the δ-perturbation-privacy, the adversary may learn additional information. For each observed link (u, w) in G^*, the adversary learns that w belongs to $Dst(u) \cup DDS(u, r, s)$, but does not know whether $w \in Dst(u)$ or $w \in DDS(u, r, s)$ because neither $Dst(u)$ nor $DDS(u, r, s)$ is published. Note that $DDS(u, r, s)$ contains at least as many nodes as $Dst(u)$ (Theorem 1). In Case 1 for $DDS(u, r, s)$, the adversary learns that w is a neighbor in $N_r(u) - N_1(u)$ $(r \geq 2)$, whereas in the other cases for $DDS(u, r, s)$ the adversary learns even less because w may be at a larger distance from u. Learning such "indirect neighbors" does not conflict with our link privacy requirement that aims to hide a direct link between two nodes. The data publisher can always use a larger r to reduce the information of such learning, at the cost of more structural distortion.

So far, we have not considered adversary's background knowledge. The most common background knowledge is the degree of a node. In the extreme case of a small graph, background knowledge on the degree of a node may be applied to infer a true link. To illustrate this, consider a tiny graph $G(V, E)$ with $V = \{u, v, w\}$ and $E = \{(u, v), (v, w)\}$. Note $Src(G) = Src(G^*)$ and that the adversary always knows the out-degree of a node in G from G^* because our algorithm never modifies the source of a link. Suppose that the link (u, v) is randomized to (u, w) and the link (v, w) remains unchanged. Suppose that the adversary has the background knowledge that the in-degree of v and w in G is 1. Then based on G^* the adversary learns that either (u, w) or (v, w) is a false link because the in-degree of w in G^* is 2. Since the in-degree of v in G^* is 0 and u is the *only* possible source for v in G^*, he learns that (u, v) must be a link in G.

The above disclosure is due to the fact learnt from G^* that u is the only candidate source for v because $Src(G^*)$ contains only two nodes, one of them being v. However, for a real life social network, G is much larger and is highly sparse where $|Src(G)|$ is much larger than the maximum in-degree of any destination node, so the adversary will face more uncertainties about the source nodes of a destination. We note that our randomization operator does not alter the out-degree of a source node. To prevent potential re-identification of a source node by the background knowledge on out-degree, prior to applying our link perturbation, degree anonymization such as [19] could be first applied to G to achieve k-anonymity (for some small $k > 1$) on the out-degree of source nodes. However, our link perturbation does alter the in-degree of a destination node in a non-deterministic manner, thus, renders the background knowledge on in-degree less effective. If necessary, degree anonymization can also be applied to anonymize the in-degree of a node prior to applying our link perturbation.

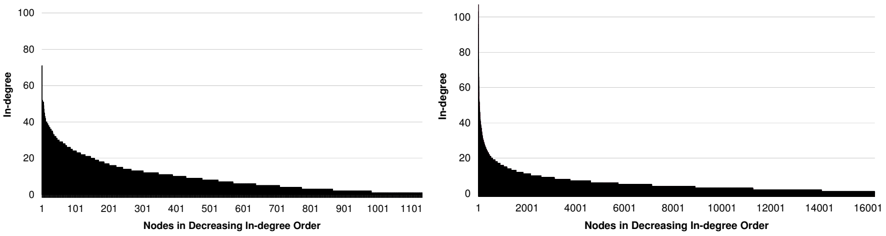
5 Experiments

In this section, we evaluate the proposed neighborhood randomization algorithm. The implementation was in VC++ on a system with core-2 Duo 2.99GHz CPU and 3.83 GB RAM.

5.1 Experiment Setup

5.1.1 Data Sets

We evaluate our proposed method on two real life social network data sets which vary in terms of domain, size, and density, i.e., the number of edges over the possible number of edges. Since social networks are generally following a similar power-law degree distribution pattern [4], we believe that the results on these data sets can be similarly generalized to other social networks as well.



(a) *URV* email network (b) *Newman*'s co-authorship network

Fig. 2 Networks long-tail degree distribution

The first data set is the e-mail network of *University Rovirai Virgili* with 1,132 nodes and 10,900 links (max in-degree 71) [12]. Links represent e-mails between university members. The data provider has eliminated e-mails sent to more than 50 different recipients (i.e., spam ignored) and also only considered bidirectional interchanges. The graph density of this network is 0.008.

The second data set is the *Newman*'s co-authorship network [23] with 16,264 nodes and 95,188 links (max in-degree 107), which is a co-authorship network of scientists posting on the High-Energy Theory in *arXiv* E-Print Archive. Bidirectional links between authors indicates their co-authorship. The graph density of this network is 0.0003.

Figure 2 shows the power-law (long-tail) degree distribution of these data sets. Note that the co-authorship network is about 15 times larger and 27 times sparser than the e-mail network.

5.1.2 Evaluation Methods

We compare the following approaches on preservation of utility metrics for social network analysis. These approaches apply a similar randomization technique to limit the link disclosure in published social networks, with a privacy notion adaptable to ours.

***Neighborhood Randomization* (NR)**: This is the proposed method in Algorithm 1. This method employs the privacy parameter δ, the neighborhood radius r, and the size $s(u)$ for $DDS(u, r, s)$ for a source u. We set r to 2, 3, 4 and 5 and set $s(u)$ to $2\times|Dst(u)|$, $3\times|Dst(u)|$, and $4\times|Dst(u)|$ (denoted by $2\times$, $3\times$, and $4\times$ in figures). The conditions in Corollary 1 are satisfied for all these settings on the two data sets.

***Graph-Wise Randomization* (GR)**: This is the special case of NR when $DSS(u, r, s)$ contains all destinations except for those in $N_1(u)$, i.e., the true destinations and u itself. Therefore, this method randomizes the destination of a link without considering structural proximity.

***Random Add/Del* (RAD)**: This is the traditional graph-wise link perturbation in [26], which randomly adds n non-existing links and randomly deletes n existing links. In this method, both the link retention probability and the probability $Pr[a_{ij}{=}1|a^*_{ij}{=}1]$ are equal to $\frac{|E|-n}{|E|}$, i.e., the fraction of true links in the sanitized graph. δ-perturbation-privacy implies $Pr[a_{ij}{=}1|a^*_{ij}{=}1] \leq 1-\delta$, i.e., $\frac{|E|-n}{|E|} \leq 1-\delta$. The minimum n satisfying this condition is $n{=}\delta{\times}|E|$.

Subgraph-wise Perturbation **(SP)**: This is the subgraph-wise perturbation in [21]. The idea of SP is to preserve graph structure by partitioning the graph into subgraphs and randomizing the destinations within each subgraph using graph-wise randomization. The work in [21] ensures a notion of (ρ_1, ρ_2)-privacy [9], where $0 < \rho_1 < \rho_2 < 1$, which states that if the prior belief about the destination of a link in G is bounded by ρ_1, the posterior belief, given the published graph G^*, is no more than ρ_2. δ-perturbation-privacy implies that $\rho_2 = 1 - \delta$.

We evaluate these methods in terms of preservation of vital metrics for social network analysis (SNA). At the graph level, we consider *average shortest distance* and *largest eigenvalue*. At the node level, we consider *degree centrality*, *closeness centrality*, *betweenness centrality*, *transitivity*, and *PageRank* (see definitions in [24][6]). A method is preferred if these metrics of the sanitized graph are similar to those of the original graph. All SNA metrics are computed by the *UCINET* software [5], and PageRank is computed by our own implementation. All results are the average of 10 runs of a randomization method.

5.2 The Findings

Sections 5.2.1 and 5.2.2 present the findings on graph level metrics and on node level metrics, respectively. The privacy level δ for δ-perturbation-privacy is set to 0.5. Section 5.2.3 presents the findings on the trade-off between privacy and utility by considering various settings of δ. Section 5.2.4 compares the recent method NR with SP.

Our goal is not to show that the chosen parameter values are better than others. Instead, our evaluation focuses on how the choices of these parameters affect privacy and utility. Ultimately which choice is better is up to the user. If the user cares more about privacy, then a large δ (small retention probability) and large $DDS(u, r, s)$ will be better because they provide more privacy protection but at the cost of less utility. In contrast, if the user is more demanding on utility, then a small δ and $DDS(u, r, s)$ is preferred, but more privacy will be lost.

5.2.1 Graph Level SNA Metrics

For the metrics at the graph level, we measure the *relative error* $\frac{\mu-\mu^*}{\mu}$, where μ and μ^* are the values of a metric in the original graph and the sanitized graph, respectively.

Average Shortest Distance. The shortest distances between nodes in a social network have applications in message spreading, searching, and calculation of SNA metrics such as centrality. The positive relative error of average

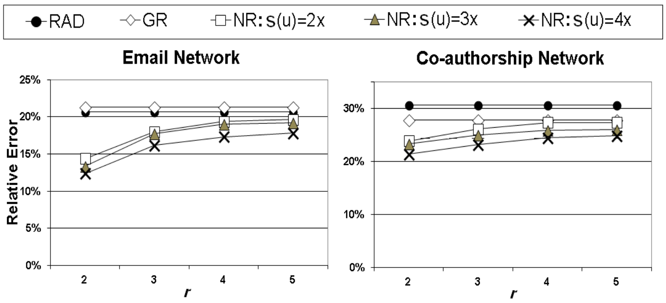


Fig. 3 Average shortest path comparison

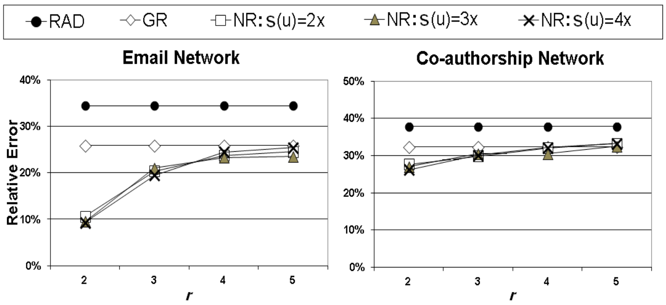


Fig. 4 Largest eigenvalue comparison

shortest distances between all pairs of nodes in Figure 3 suggests that link perturbation tends to reduce the average shortest distance. To understand why, recall that social networks follow the power-law degree distribution, that is, a small number of nodes have high in-degree and a majority of nodes have low in-degree (as in Figure 2). Due to this property, a *randomly selected* false destination is more likely to be a low in-degree node, as such, randomization tends to reduce the in-degree of a high in-degree node and increase the in-degree of a low in-degree node. Consequently, the difference in in-degree is reduced by link randomization. We refer to this phenomenon as "*degree smoothing effect*".

The degree smoothing effect increases the in-degree of many nodes, thus, causes more nodes being on shortest paths (higher degree nodes are more likely to be on shortest paths), which leads to a reduction in the average shortest distance. However, this effect is less prevalent for NR where the random selection of false destinations is limited to a small neighborhood, in which case fewer low in-degree nodes will have their in-degree increased. This explains why the relative error of NR is smaller than those of RAD and GR. We note that $s(u)$ has only a limited effect on the relative error, though a larger $s(u)$ provides more choices of a false destination.

Largest Eigenvalue. A non-zero vector $\overrightarrow{v}$ is an eigenvector of a square matrix A if there is a scalar λ such that $A\overrightarrow{v} = \lambda\overrightarrow{v}$. λ is called the eigenvalue for $\overrightarrow{v}$. There may be many pairs of eigenvectors and eigenvalues. The set of eigenvalues of the adjacency matrix of a graph defines the *spectrum* of that graph and has a close relation to many graph characteristics. It was shown that *maximum degree*, *chromatic number*, *clique number*, *epidemic threshold* of virus propagation are all related to the largest eigenvalue [26]. The above mentioned degree smoothing effect is also reflected on eigenvalues, which results in reduction in the largest eigenvalue. Since NR has less degree smoothing effect than GR and RAD, NR better preserves the largest eigenvalue than GR and RAD, as shown by the smaller relative error in Figure 4.

5.2.2 Node Level SNA Metrics

For a metric at the node level, preserving the relative rank of nodes is more important than having a small error of the metric, and higher ranked nodes have more weight than lower ranked nodes in this preservation. For example, for pagerank, only pages near to the top of the ranked list are interesting, so preservation of top ranked pages is far more important than preservation of lower ranked pages. For this reason, we evaluate the *Spearman similarity* [10] between the ranked list of nodes L computed using the original graph and the ranked list of nodes L^* computed using the sanitized graph. Let $r_L(x)$ be the rank of x in the list L. The Spearman similarity of the top k nodes in L and L^* is defined as $1-d$, where d is the *Spearman distance* $d = \frac{2(k-|Z|)(k+1)+A-B-C}{k(k+1)}$, where $A = \sum_{i \in Z} |r_L(i) - r_{L^*}(i)|$, Z is the set of nodes in the top k sublist of both L and L^*, $B = \sum_{i \in S} r_L(i)$, S is the set of nodes in the top k sublist of only L, $C = \sum_{i \in T} r_{L^*}(i)$, and T is the set of nodes in the top k sublist of only L^*. Note that Spearman similarity ranges from 0 (totally reversed) to 1 (totally identical) [10]. We report the (Spearman) similarity of top 50% nodes. The results of evaluating other top percentages are not very different.

Degree Centrality. This is defined as the degree of a node v. A node with a higher degree centrality is generally a more active player in the network. For a directed network, in-degree is interpreted as a form of popularity. We focus on the in-degree since NR and GR never alter the out-degree of a node. Figure 5 shows the Spearman similarity for degree centrality of nodes with the original graph being the baseline. The similarity for NR is significantly higher that for GR and RAD, especially for a small r. This is because NR chooses the randomized destination of a link from a local neighborhood, thus, suffers from less degree smoothing effect than RAD and GR, as discussed before. Therefore, NR incurs less changes in the in-degree rank of nodes.

Betweenness Centrality. This is defined as the number of times that a node occurs on geodesics (i.e., shortest paths) linking other nodes in a graph [24]. Formally, the betweenness centrality of a node v is $\sum_{s,t \in V} \sigma_{st}(v)$, where s and t are not v and $\sigma_{st}(v) = 1$ if the shortest path from s to t passes through v and 0 otherwise. The betweenness centrality is a measure of information control in a network. A node with a high betweenness centrality generally has more influence on data flow in a network. In general, nodes with a higher in-degree are more likely to have more shortest paths crossing them, thus, a higher betweenness centrality. Therefore, the degree smoothing effect of link

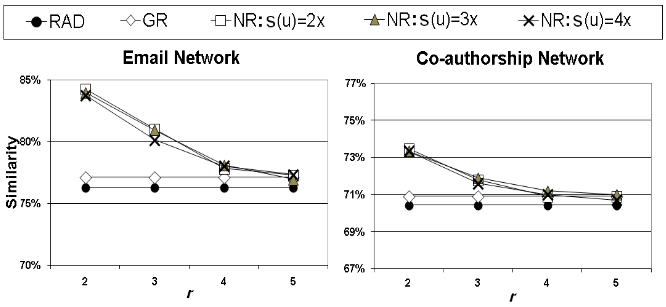


Fig. 5 Degree centrality comparison

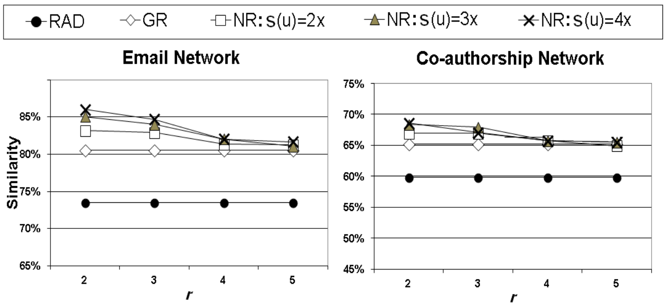


Fig. 6 Betweenness centrality comparison

perturbation translates into an increase in betweenness centrality for low in-degree nodes and a decrease in betweenness centrality for high in-degree nodes. Since the degree smoothing effect is less for NR than for RAD and GR, NR has a higher similarity than RAD and GR. This is confirmed by the results in Figure 6.

Closeness Centrality. The closeness centrality of a node v is the reciprocal of the sum of geodesic distances to other nodes reachable from v [24], defined as $\frac{1}{\sum_{t \in V} d(v,t)}$, where $d(v,t)$ is the shortest distance from v to t and $d(v,t) \neq \infty$. It measures the accessibility of v to other nodes in a network, and a node with a high closeness centrality is generally close to other nodes and can spread information to others faster. As noted in Section 4.3.2, increasing r results in decreasing the average shortest path length, thus, increasing the closeness measure. This results in decreased similarity as depicted in Figure 7. Again, with less degree smoothing effect NR is able to better preserve the ranking of closeness centrality than RAD and GR, especially for a small r.

Transitivity. The transitivity (a.k.a local clustering coefficient) of a node measures the extent to which neighbor nodes of the node are connected [24]. For example, a node whose friends are also friends with each other has a high transitivity. In a social network, people in a small group (where nodes have a low degree) are usually connected together, whereas people connected to celebrity nodes (nodes with a high degree) might not be well connected them-

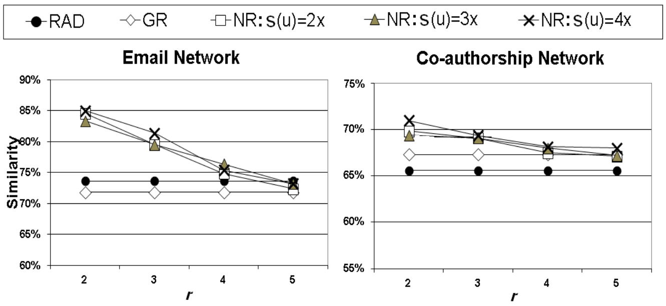


Fig. 7 Closeness centrality comparison

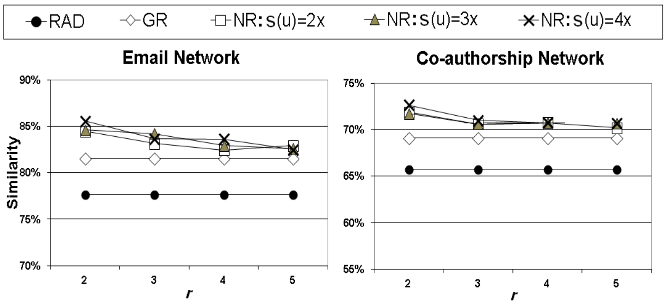


Fig. 8 Transitivity comparison

selves. Therefore, nodes with a lower degree tend to have a higher transitivity. Figures 8 shows the similarity for transitivity. Recall the finding in Section 5.2.1 that RAD and GR have a more degree smoothing effect (compared to NR), thus, more changes in in-degree, and more changes in the transitivity ranking of nodes. This explains why RAD and GR have a smaller similarity than NR in Figure 8.

PageRank. A variant of eigenvector centrality is employed by Google's *PageRank* link analysis for web pages [6]. The PageRank for a page u can be expressed as $PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)}$, where the set B_u contains all pages linking to u and $L(v)$ is the number of links from the page v. $PR(u)$ indicates the likelihood of arriving at u by a random surfer and can be considered as a measure of relative importance of u in the Web. For a social network, u represents a node instead of a web page and $PR(u)$ is the popularity of u. The study in [18] suggested that the distribution of PageRank is related to the distribution of in-degree. Therefore, the degree smoothing effect of link perturbation in Section 5.2.1 would have a similar smoothing effect on PageRank of nodes. As we observed there, NR incurs less smoothing effect than GR and RAD. Consequently, PageRank is better preserved in NR than in GR and RAD. This observation is confirmed by the higher similarity of NR in Figure 9.

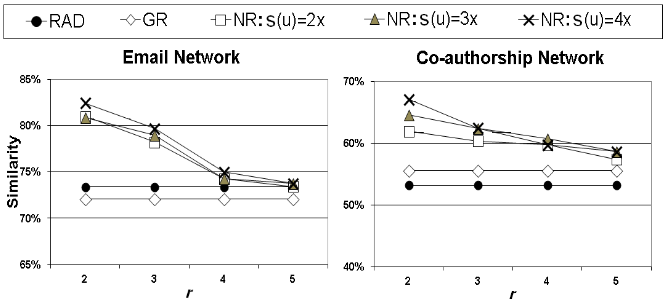


Fig. 9 PageRank comparison

5.2.3 Privacy vs. Utility

In this experiment, we focus on NR and study the effect of the privacy parameter δ (for δ-perturbation-privacy) on all SNA metrics. A larger δ means a smaller retention probability p (which is equal to $1-\delta$) of perturbing a link, thus, more uncertainty in inferring a true link in the sanitized graph. We set δ to 0.3, 0.5, and 0.7, and set $r = 2$ and $s(u)$=2×$|Dst(u)|$. Figure 10 shows that a larger δ leads to a reduced similarity and a higher relative error for all metrics considered. This finding is expected in that a higher privacy level incurs a higher price paid for utility.

5.2.4 Comparison with Subgraph-Wise Perturbation

Finally, we compare NR with SP [21]. SP shares a similar motivation with NR, i.e., preserving structural proximity by limiting the randomization domain for the destination of a link to subgraphs. Specifically, it partitions the graph into some number of (small) subgraphs and performs link destination randomization within subgraphs independently. SP employs the (ρ_1, ρ_2)-privacy [9], which informally says that if the adversary's *prior* belief (before seeing the published graph) that a node v is the destination of a link is no more than ρ_1, his *posterior* belief (after seeing the perturbed graph) that v is the true destination of a link is no more than ρ_2, where $0 < \rho_1 < \rho_2 < 1$. For SP, we set ρ_1=0.01, $\rho_2 = 0.5$, and the number of subgraphs to 10 and 50 for the email network and 100 and 500 for the larger co-authorship network. For NR, we set $\delta = 0.5$ (because $\rho_2 = 0.5$), $r = 2$ and $s(u)$=2×$|Dst(u)|$.

Figure 11 shows that the similarity and the relative error of NR is better than SP with smaller number of subgraphs and slightly worse than SP with larger number of subgraphs. However, as explained in [21], SP introduces new threats of identifying a true link, as the number of subgraphs increases and each subgraph becomes small. In particular, a node may become a popular destination in a small subgraph even though it is not so in the entire graph. This clearly increases the chance of such nodes being re-identified as the destination of a link in the subgraph. In the terminology of ρ_1-ρ_2 privacy, this

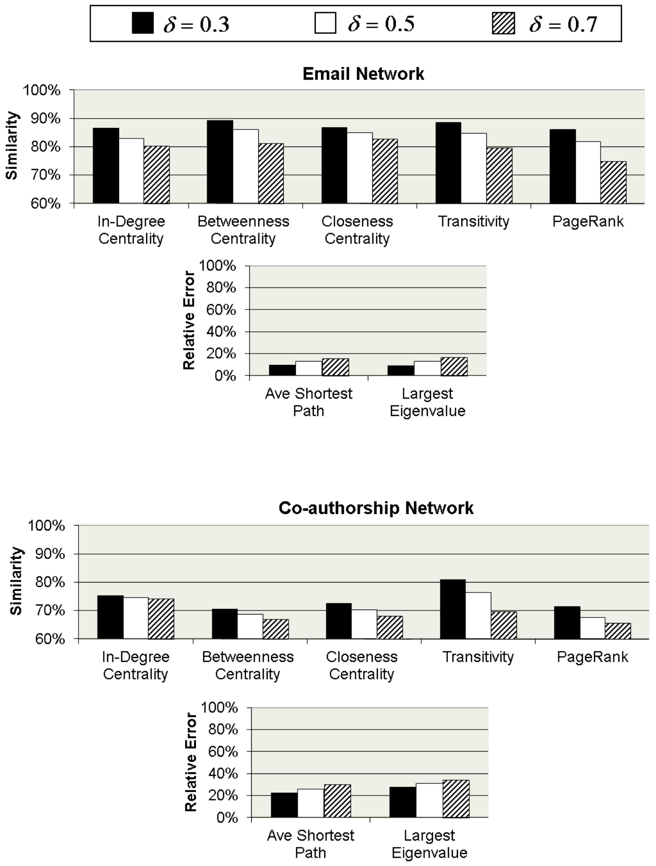


Fig. 10 Privacy vs. utility for social network analysis ($r = 2$, $s(u) = 2 \times |Dst(u)|$)

change of popularity could cause the prior belief of such nodes as destinations in the subgraph to exceed ρ_1, in which case such nodes are no longer protected by the randomization in the subgraph.

Indeed, the study in [21] shows that for the URV data set, only 41.53% of the nodes protected in the original graph remained protected in a subgraph when G is partitioned into 10 subgraphs, and this percentage reduces to 11.98% and 4.85% when G is partitioned into 50 and 100 subgraphs, respectively. To tackle this issue, an additional step, called degree balancing, was used in [21] to heuristically move links between subgraphs. However, as shown in [21], this step will introduce additional structural distortion which goes against the idea of graph partitioning. The degree balancing procedure, however, does not guarantee to always eliminate the above threat. We turn off the degree balancing process in our experiments and thus the result of SP in Figure 11 has not considered the additional structural distortion. NR, on the other hand, does not suffer from this drawback because it does not partition the graph.

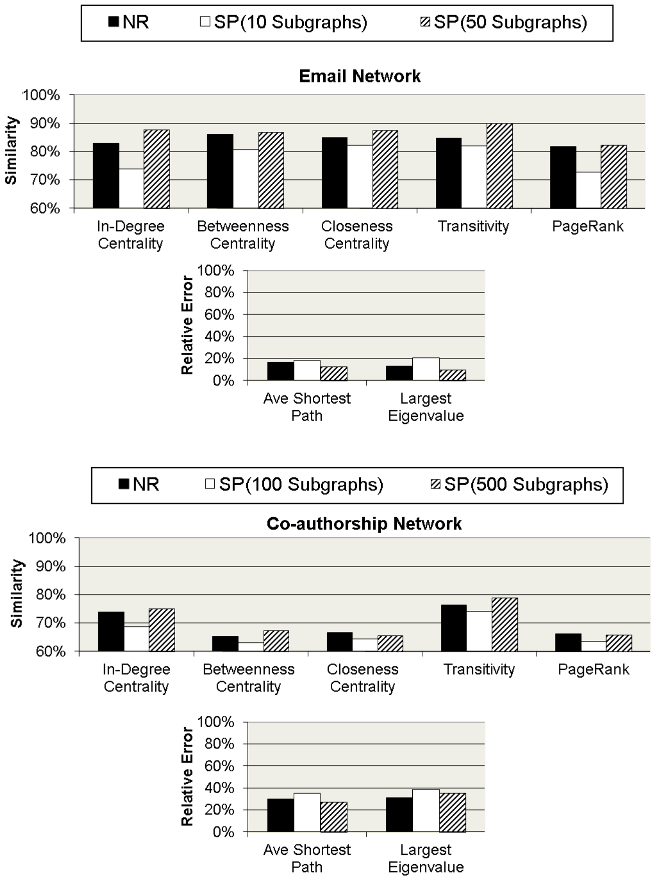


Fig. 11 NR ($r = 2$, $s(u) = 2 \times |Dst(u)|$, $\delta = 0.5$) vs. SP (ρ_1=0.01, $\rho_2 = 0.5$)

5.3 Summary of Findings and Discussion

Our study shows that the proposed neighborhood randomization indeed better preserves vital information of social networks than previous link perturbation methods. Results of our experiment (in figures 4 and 5) show an average of of about 35% improvement for NR in reducing the relative error of graph level SNA metrics compared to RAD and GR. There is also an average improvement (shown in figures 5 to 9) of about 10% in similarity for node level SNA metrics compared to RAD and GR. Note that these improvements over many settings are indeed significant. This is a consequence of the structure-aware randomization of the destination of a link. The radius of the neighborhood for randomization (i.e., r) plays an important role in this preservation whereas the size of the neighborhood (i.e., s) has only limited effect.

Both *neighborhood randomization* and *subgraph-wise perturbation* [21] better preserve the structural proximity compared to traditional work because the randomization is limited to local neighborhoods or subgraphs. These methods preserve the out-degree of nodes in the published graph, however, the *subgraph-wise perturbation* also enables the data publisher to reconstruct in-degree of nodes using iterative Bayesian reconstruction [1]. To do so, the data publisher should also provide researchers with the destination randomization domain, i.e., publishing each subgraph separately. The in-degree of nodes has applications such as popularity based ranking and influence of nodes.

The major problem with *subgraph-wise perturbation* is that with the growth of number of subgraphs, a node may become a popular destination in a small subgraph even though it is not so in the entire graph. This popularity change can make that node no longer protected with respect to the ρ_1-ρ_2 privacy by the randomization in the subgraph. Although the degree balancing process [21] helps partially to reduce such threats, it introduces additional structural distortion which goes against the idea of graph partitioning. *Neighborhood randomization*, on the other hand, does not suffer from this drawback because it does not partition the graph.

Although more data sets are always better for evaluation, the two data sets used were carefully selected in terms of varied domain, size, and density. The similar utility improvement observed in both network data suggests that the proposed neighborhood randomization method is a promising one. Like many works in the literature in a similar nature, we have no intent to claim that our method definitely" outperforms previous work. Instead, we rely on the intuition of the proposed neighborhood randomization as a heuristic for better results, and we believe that this heuristic works most of the time.

6 Conclusion

Link perturbation is a powerful technique for preserving link privacy while allowing social network analysis. The standard link perturbation causes significant structural distortion due to insensitivity to structural proximity. In this work, we presented a novel structure-aware link perturbation scheme, neighborhood randomization. This scheme preserves more graph structures through perturbing only the destination of a link and limiting the randomized destination to a close neighborhood. Two main contributions are the formulation of "close neighborhood" that satisfies certain essential requirements, and the study on the effectiveness of this approach in preserving graph structures. Our studies confirmed that neighborhood randomization better preserves vital information for social network analysis than previous link perturbation techniques.

Acknowledgements Ke Wang's work is partially funded by a Discovery Grant from Natural Sciences and Engineering Research Council of Canada and is partially done when Ke Wang visited SA Center for Big Data Research hosted in Renmin University of China. This Center is partially funded by a Chinese National "111" Project "Attracting International Talents in Data Engineering and Knowledge Engineering Research".

References

1. Agrawal, R., Srikant, R., Thomas, D.: Privacy preserving OLAP. In SIGMOD (2005)
2. Backstrom, L., Dwork, C., Kleinberg, J. M.: Wherefore art thou R3579X?: anonymized social networks, hidden patterns, and structural steganography. In WWW (2007)
3. Bai, K., Liu, Y., Liu, P.: Prevent identity disclosure in social network data study. In ACM CCS (2009)
4. Barabasi, A.-L., Albert, R.: Emergence of scaling in random networks. In Science, V. 286, 509–512, (1999)
5. Borgatti, S.P., Everett, M.G., Freeman, L.C.: Ucinet for windows: software for social network analysis. Harvard, MA: Analytic Technologies (2002)
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. In WWW (1998)
7. Campan, A., Truta, T.: A clustering approach for data and structural anonymity in social networks. In PinKDD, (2008)
8. Cheng, J., Fu, A. W., Liu, J.: K-isomorphism: privacy preserving network publication against structural attacks. In SIGMOD (2010)
9. Evfimievski, A., Gehrke, J., Srikant, R.: Limiting privacy breaches in privacy preserving data mining. In PODS (2003)
10. R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. SIAM Jour. on Discrete Math., 17(1), (2003)
11. Fung, B. C. M., Wang, K., Fu, A. W.-C., Yu, P. S.: Introduction to privacy-preserving data publishing: concepts and techniques. Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC, (2010)
12. Guimera, R., Danon, L., Guilera, A., Giralt, F., Arenas, A.: Self-similar community structure in a network of human interactions. In Physical Review, V. 68, (2003)
13. Hay, M., Miklau, G., Jensen, D., Weis, P., Srivastava, S.: Anonymizing social networks. Technical report, University of Massachusetts Amherst, (2007)
14. Hay, M., Miklau, G., Jensen, D., Towsley, D., and Weis, P.: Resisting structural reidentification in anonymized social networks. In VLDB (2008)
15. He, X. Vaidya, J., Shafiq, B., Adam, N., Lin, X.: Reachability Analysis in Privacy-Preserving Perturbed Graphs. In WI-IAT (2010)
16. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. In SIAM Journal on Scientific Computing, Vol. 20, (1999)
17. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In CIKM (2003)
18. Litvak, N., Scheinhardt, W. R. W., Volkovich, Y.: In-Degree and PageRank: Why do they follow similar power laws? In Internet Mathematics 4(2), (2007)
19. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In ACM SIGMOD/PODS (2008)
20. Medforth, N., Wang, K.: Privacy risk in graph stream publishing for social network data. In ICDM (2011)
21. Milani Fard, A., Wang, K., Yu, P.S.: Limiting link disclosure in social network analysis through subgraph-wise perturbation. In EDBT (2012)
22. Musiał, K., Kazienko, P.: Social networks on the Internet. World Wide Web Journal, Springer, V. 16, N. 1, 31–72, (2013)
23. Newman, M. E. J.: The structure of scientific collaboration networks. In Proc. of the National Academy of Sciences of the USA,V. 98, N. 2, (2001)
24. Wasserman, S., Faust, K.: Social network analysis: methods and applications. Cambridge University Press, (1999)
25. Wong, R. C., Fu, A. W., Wang, K., Pei, J.: Minimality attack in privacy preserving data publishing. In VLDB (2007)
26. Ying, X., Wu, X.: Randomizing social networks: a spectrum preserving approach. In SDM (2008)
27. Ying, X., Wu, X.: On link privacy in randomizing social networks. In PAKDD (2009)
28. Zhang, L., Zhang, W.: Edge anonymity in social network graphs. In IEEE Social Computing (2009)

29. Zheleva, E., Getoor, L.: Preserving the privacy of sensitive relationships in graph data. In PinKDD (2007)
30. Zhou, D., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In ICDE (2008)
31. Zou, L., Chen, L., Ozsu, M. T.: K-automorphism: a general framework for privacy preserving network publication. In VLDB (2009)