Estimating Properties of Social Networks via Random Walk considering Private Nodes

Kazuki Nakajima
Tokyo Institute of Technology
nakajima.k.an@m.titech.ac.jp

Kazuyuki Shudo
Tokyo Institute of Technology
shudo@is.titech.ac.jp

ABSTRACT

Accurately analyzing graph properties of social networks is a challenging task because of access limitations to the graph data. To address this challenge, several algorithms to obtain unbiased estimates of properties from few samples via a random walk have been studied. However, existing algorithms do not consider *private nodes* who hide their neighbors in real social networks, leading to some practical problems. Here we design random walk-based algorithms to accurately estimate properties without any problems caused by private nodes. First, we design a random walk-based sampling algorithm that comprises the neighbor selection to obtain samples having the Markov property and the calculation of weights for each sample to correct the sampling bias. Further, for two graph property estimators, we propose the weighting methods to reduce not only the sampling bias but also estimation errors due to private nodes. The proposed algorithms improve the estimation accuracy of the existing algorithms by up to 92.6% on real-world datasets.

CCS CONCEPTS

• **General and reference → Estimation**; • **Mathematics of computing → Graph algorithms**.

KEYWORDS

Social networks; Estimation; Sampling; Random walk; Private nodes

ACM Reference Format:
Kazuki Nakajima and Kazuyuki Shudo. 2020. Estimating Properties of Social Networks via Random Walk considering Private Nodes. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403116

1 INTRODUCTION

Online social networks (OSNs) have been primarily studied to understand the nature of global social structures, such as human connections and behaviors [1, 9, 10, 16, 20, 29]. A widely used approach to understand the structure is to analyze various graph properties, such as the number of nodes (size), the average degree, and the degree distribution. However, an accurate analysis of social graphs is a challenging task because of access limitations to the graph data. Although some studies used complete graph data [1, 3, 21], all the data is typically not available to researchers. In practical scenarios, we sample a part of graph data through the application programming interfaces (APIs) to estimate properties [9, 10]. We can obtain unbiased estimates if users are randomly sampled; however, it is frequently difficult to randomly assign user IDs to the APIs [6, 11].

Several algorithms [5, 6, 11, 12, 22, 28] based on the *re-weighted random walk* scheme [9, 10] have been studied to address this challenge. Many OSNs have provided APIs to return the neighbor data of a queried user. By repeatedly utilizing this function for a randomly selected neighbor, we obtain samples with the Markov property via a random walk on a social network. Then, we obtain unbiased estimates of properties by re-weighting each sample to correct the sampling bias derived from the Markov chain analysis. Previous studies have focused on accurate estimates with a small number of queries because *APIs are typically rate-limited* [6, 11].

However, previous studies have ignored *private users* who do not provide their neighbor data even if queried. Although previous studies assume a network comprising only *public users* who publish their neighbor data, a certain percentage of private users exist in real social networks. For example, private users were about 27% on Facebook [4] and 34% on Pokec [25], which is an OSN in Slovakia.

Private users cause some problems for existing algorithms. For example, how do we deal with private users in a random walk sampling? If private users are sampled, we need to handle an exception wherein the neighbor data cannot be obtained by approaches such as returning to the public user sampled previously. However, if such an exceptional process is performed, the Markov property of a sample sequence cannot be guaranteed. This can prevent us from obtaining unbiased estimates of properties. There is another serious problem. A fast solution of problems in sampling is to not include private users to the sample sequence; however, by not sampling them, the sampling bias and weighting for each sample are different from the assumptions of existing algorithms. Thus, the existing algorithms cause estimation errors due to private users.

We aim to design re-weighted random walk algorithms considering private users to accurately estimate properties without any problems caused by private users. To the best of our knowledge, this study is the first to focus on the effect of private users on random walk-based estimators. First, we make three assumptions and define two access models (*the ideal model* and *the hidden privacy model*), based on previous studies and our observations of real social networks (Section 2). Then, we design sampling and estimation algorithms based on these assumptions and access models.

This study has two main contributions. The first contribution is to enable us to *successfully perform re-weighted random walk-based algorithms in real social networks including private users* (Section 3). We discuss the neighbor selection in a random walk by considering

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08...$15.00
https://doi.org/10.1145/3394486.3403116

Proc. 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining
(KDD 2020), August 2020

private users and derive the sampling bias of each user. Then, for each access model, we describe the calculation of the weights to correct the sampling bias; particularly for the hidden privacy model, we propose a method to approximate weights with much fewer queries than exact calculation.

The second contribution is to enable us to *accurately estimate the size and average degree of a whole social graph including private nodes* (Section 4). Existing algorithms [6, 11, 12] cause errors due to private users because the conventional weighting aims only to correct the sampling bias. We propose weighting methods to reduce both sampling bias and errors due to private users. Moreover, we theoretically explain that estimates obtained using the proposed weighting converge to values with smaller expected errors than the previous weighting.

Finally, we validate the theoretical claims and effectiveness of the proposed algorithms on extensive experiments using real social network datasets (Section 5). It is important to evaluate the performance when the assumptions are not satisfied because the proposed algorithms are designed on some simplified assumptions. We empirically show that the proposed algorithms perform well on realistic datasets, such as a 92.6% improvement in the estimation accuracy on the Pokec dataset including real private users.

2 PRELIMINARIES

2.1 Definitions and Notations

We represent a social network as a connected and undirected graph, $G = (V, E)$, where $V = \{v_1, ..., v_n\}$ denotes the set of nodes (users), and E denotes the set of edges (friendship). Let d_i denote the degree (number of neighbors) of v_i and $D = \sum_{v_i \in V} d_i$ denote the sum of degrees. We define the average degree of G as $d_{avg} = \frac{D}{n}$. Each node, v_i, has a privacy label, $l_{pri}(i) \in \{public, private\}$. The set of privacy labels is denoted by $\mathcal{L}_{pri} = \{l_{pri}(i)\}_{v_i \in V}$.

We refer to connected subgraphs that consist of public nodes on G as *public-clusters*. Let $\{C_i\}$ denote a set of public-clusters and $C^* = (V^*, E^*)$ denote the largest public-cluster. Let n^* denote the number of nodes in C^*, and $d_i^* = |\{v_j \in V^* | (v_i, v_j) \in E^*\}|$ denote the *public-degree* (the number of neighbors that publish their own neighbors) of a node $v_i \in V^*$. Let $D^* = \sum_{v_i \in V^*} d_i^*$ denote the sum of public-degrees. We define the average degree of C^* as $d_{avg}^* = \frac{D^*}{n^*}$. Let $\mathbf{1}_{V^*}(v_i)$ denote a function that returns 1 if $v_i \in V^*$, otherwise 0.

Example: Let $v_i = i$ $(1 \leq i \leq 10)$ in Figure 1. There are three public-clusters, C_1, C_2 and C_3, and C_1 is the largest public-cluster:

- $C_1 = (\{4, 5, 6, 7, 9\}, \{(4, 5), (5, 6), (5, 7), (5, 9)\})$
- $C_2 = (\{8, 10\}, \{(8, 10)\})$
- $C_3 = (\{3\}, \{\})$.

Also, it holds $n^* = 5, d_4^* = 1, d_5^* = 4, d_6^* = 1, d_7^* = 1, d_9^* = 1$.

2.2 Assumptions

We design algorithms based on the following three assumptions.

(1) *The indices of all neighbors, including private neighbors, of a queried public node are obtained.* Facebook as of the previous study [10] satisfied this assumption. We have confirmed that Twitter as of June 2020 satisfies this assumption [26, 27].

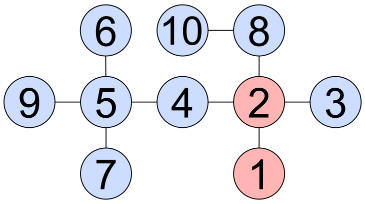


Figure 1: An example of a graph with privacy labels. Here, nodes 1 and 2 are private and others are public.

(2) *Each node independently becomes private with probability p, otherwise public, where* $0 \leq p \leq 1$. We assume that each node independently hides neighbors with a certain probability[1].

(3) *The seed of a random walk is on the largest public-cluster.* We do not consider the number of queries performed to select the seed from nodes on the largest public-cluster, because we consider that this number is sufficiently small.

2.3 Access Models

We do not assume the acquisition of all the graph data and their random access. We assume that a graph G is static.

If we query a public node, the neighbor data are provided. If a private node is queried, the neighbor data cannot be obtained[2]. We consider two models for the neighbor data that are provided when a public node, v_i, is queried: ideal model and hidden privacy model.

The ideal model: *Indices (IDs) and privacy labels of all neighbors of v_i are obtained.* For example, when a public node 4 is queried in Figure 1, we obtain the set $\{(2, private), (5, public)\}$. Facebook as of the previous study [10] corresponds to this access model.

The hidden privacy model: *Even though ids of all neighbors of v_i are obtained, the privacy labels are not.* For example, when a public node 4 is queried in Figure 1, we obtain the set $\{2, 5\}$. Twitter API as of June 2020 corresponds to this model [26, 27].

2.4 Markov Chain Basics

We introduce the basics of a Markov chain for theoretical analysis of random walk-based estimators. First, we describe the stationary distribution of a Markov chain, which is needed to derive the sampling bias. Let $\boldsymbol{P} = \{P_{i,j}\}_{i,j \in S}$ denote the transition probability matrix of a Markov chain on the state space, S. If $\pi_j = \sum_{i \in S} \pi_i P_{i,j}$ holds for all $j \in S$, $\boldsymbol{\pi} = \{\pi_i\}_{i \in S}$ is the stationary distribution of $\boldsymbol{P}$. The following theorem holds for the stationary distribution $\boldsymbol{\pi}$ of $\boldsymbol{P}$:

THEOREM 2.1. *[19] If $\boldsymbol{P}$ is ergodic (i.e., irreducible and aperiodic), the stationary distribution, $\boldsymbol{\pi}$, uniquely exists.*

Then, we review the strong law of large numbers for a Markov chain, which is needed to ensure that the sample average converges almost surely to its expected value:

THEOREM 2.2. *[5, 17] Let $\{X_k\}$ be an ergodic Markov chain with the stationary distribution $\boldsymbol{\pi}$ on the finite state space S. For any function $f : S \rightarrow \mathbb{R}$, $\frac{1}{r} \sum_{k=1}^{r} f(X_k)$ converges to $\sum_{i \in S} \pi_i f(i)$ almost surely as $r \rightarrow \infty$ regardless of the initial distribution of the chain.*

[1] Intuitively, private nodes tend to have low degrees under Assumption 2, because the degree distribution of social networks is typically biased to low degrees [1, 9, 10, 16, 20].
[2] The output may be an error message or an empty set.

3 RANDOM WALK SAMPLING

We design a random walk-based sampling algorithm that consists of the neighbor selection to obtain samples having the Markov property and the calculation of weights to correct the sampling bias. The proofs in this section are shown in the supplement.

3.1 Transition Neighbor Selection

In a simple random walk, we move to a randomly selected neighbor. If private nodes are sampled, practical problems on estimating properties occur, because their neighbor data are unclear. For example, we cannot simply continue the algorithm, and it is difficult to correct the sampling bias of sampled private nodes.

We believe that it is appropriate to move to a public neighbor randomly selected as Gjoka et al. [10] performed on Facebook[3]. We design the algorithm to obtain an index sequence of r sampled nodes, denoted by $\{x_k\}_{k=1}^r$, as follows. First, we select a seed, $v_{x_1} \in C^*$, as the first sample. Then, for $1 \leq k \leq r-1$, we randomly select a node, u, from neighbors of the k-th sample, v_{x_k}. If u is public, we move to u as next sample, $v_{x_{k+1}}$; otherwise, we randomly reselect a neighbor of v_{x_k}. In the ideal model, we check if a selected neighbor u is public without an additional query. In the hidden privacy model, we judge the privacy label of u by additionally querying u.

3.2 Sampling Bias

We derive the sampling bias caused by the designed random walk. Let the probability that an event A will occur be denoted by $Pr[A]$. We define the distribution induced by the sequence of sampled indices as $\boldsymbol{\pi}_r = (Pr[x_r = i])_{v_i \in V}$.

We show that each node on the largest public-cluster is sampled in proportion to the public-degree via the designed random walk.

Lemma 3.1. *$\boldsymbol{\pi}_r$ converges to the vector $\boldsymbol{\pi} = (p_i \mathbf{1}_{V^*}(v_i))_{v_i \in V}$ after many steps of the designed random walk, where $p_i = \frac{d_i^*}{D^*}$.*

3.3 Public-Degree Calculation

We need to calculate the public-degree of each sample to correct the sampling bias due to the public-degree.

In the ideal model, we can exactly calculate each public-degree without additional query because the privacy labels of all the neighbors are obtained. Conversely, in the hidden privacy model, the exact calculation needs a considerable number of additional queries to obtain privacy labels of all neighbors for each sample.

For the hidden privacy model, we propose a method to approximate each public-degree without additional queries by utilizing the history of neighbor selections of the designed random walk. We record two values for each sample v_{x_k}: the total number of successful public neighbor selections, a_{x_k}, and that of neighbor selections, b_{x_k}. After sampling, each approximate value, $\hat{d}_{x_k}^*$, is calculated as

$$\hat{d}_{x_k}^* \triangleq d_{x_k} \frac{a_{x_k}}{b_{x_k}}.$$

The pseudo-code of the designed random walk with the proposed approximation of public-degrees is shown in the supplement.

We ensure the convergence of the approximated public-degree after many steps of the designed random walk.

Lemma 3.2. *$\hat{d}_{x_k}^*$ converges to $d_{x_k}^*$ for each sample, v_{x_k}, after many steps of the designed random walk.*

Then, we explain that the proposed method performs much fewer queries than the exact method that exactly calculates the public-degree of each sample. Here, for simplicity, we do not consider the save of the indices of nodes queried once. $Q(k)$ and $\hat{Q}(k)$ denote the number of queries that are performed at the k-th sample by the exact and proposed methods, respectively. Let $Q = \frac{1}{r}\sum_{k=1}^r Q(k)$ and $\hat{Q} = \frac{1}{r}\sum_{k=1}^r \hat{Q}(k)$ denote the ratio of the number of queries to the sample size of the exact and proposed methods, respectively.

Lemma 3.3. *The expectations of Q and $\hat{Q}$ are $\frac{\sum_{v_i \in V^*} d_i^* d_i}{D^*}$ and $\frac{\sum_{v_i \in V^*} d_i}{D^*}$, respectively.*

Lemma 3.3 implies that the proposed method performs much fewer queries than the exact method because $\sum_{v_i \in V^*} d_i$ is intuitively much smaller than $\sum_{v_i \in V^*} d_i^* d_i$ in real social networks.

4 PROPERTIES ESTIMATION

We design estimators for the size and the average degree via a random walk designed in Section 3. We first introduce the existing estimators and then propose estimators with improved weighting. The proofs in this section are shown in the supplement.

4.1 Overview

Here we address a novel problem that the convergence values of estimators have errors caused by private nodes. The conventional estimators perform the weighting using only the public-degree to correct the sampling bias; consequently, the estimates converge to the properties of the largest public-cluster. When a graph comprises only public nodes as have been assumed in previous studies, the convergence values are equal to the properties of the original graph. However, when there are private nodes on a graph, the convergence values typically have errors due to private nodes, or more specifically, the value of p.

It is unclear how to reduce the errors due to the value of p. If we know the exact value of p, we can easily correct the errors; however, this is usually unknown. Moreover, we cannot apply existing methods to estimate the value of p based on privacy labels of samples (Section 3.C.3 in [10], Section 3.2 in [12] and Section 4.2.3 in [23]), because private nodes are not included in samples.

We propose the weighting methods combining the public-degree and degree to reduce the errors of convergence values due to the value of p. The proposed methods are based on the fact that the public-degree follows the binomial distribution with parameters of the degree and $1-p$ regarding the set of privacy labels under Assumption 2. Proposed estimators provide the same convergence values when $p = 0$ and convergence values with smaller expected errors when $p > 0$ as compared with the existing estimators.

4.2 Size Estimation

Existing estimator: The node collision (NC) algorithm is an effective estimator for the size [11, 12]. This algorithm examines sample

[3] There is no detailed discussion of the effect of private nodes on re-weighted random walk algorithms [10]. They focused on only the part comprising public nodes.

pairs whose ordinal numbers in the sample sequence are more than a threshold, m, apart. Sample pairs whose ordinal numbers in the sample sequence are sufficiently apart can be regarded as being independently sampled [11].

Let $I = \{(k, l) \mid m \leq |k - l| \wedge 1 \leq k, l \leq r\}$ denote the set of integer pairs that are between 1 and r and at least m away. Let $\phi_{k,l}$ denote a variable which returns 1 if $x_k = x_l$ (this is called a collision), and otherwise, 0. The average of the number of collisions Φ_n^{NC}, the average of the weights to correct the sampling bias Ψ_n^{NC}, and a size estimate n^{NC} obtained by NC algorithm are as follows:

$$\Phi_n^{NC} = \frac{1}{|I|} \sum_{(k,l) \in I} \phi_{k,l}, \quad \Psi_n^{NC} = \frac{1}{|I|} \sum_{(k,l) \in I} \frac{d_{x_k}^*}{d_{x_l}^*}, \quad n^{NC} \triangleq \frac{\Psi_n^{NC}}{\Phi_n^{NC}}$$

It holds the following lemma expanded from the previous studies [11, 12] which assume graph with no private nodes.

Lemma 4.1. *n^{NC} converges to size of the largest public-cluster, n^*.*

We focus on the expectation of n^* regarding the set of privacy labels, $\mathcal{L}_{pri}$, to quantify the error between n^* and n. $E_{pri}[X]$ denotes the expectation of a random variable X regarding $\mathcal{L}_{pri}$. We approximate the expectation of n^* regarding $\mathcal{L}_{pri}$ under the condition that all the public nodes belong to the largest public-cluster. Under this condition, it holds that $Pr[v_i \in V^*] = Pr[l_{pri}(i) = public] = 1 - p$.

The following lemma holds for the convergence value n^*:

Lemma 4.2. *Under the condition that all the public nodes belong to the largest public-cluster, we have*

$$E_{pri}[n^*] = (1 - p)n.$$

Lemma 4.2 implies that an expected relative error of the convergence value in the existing estimator is $1 - p$.

Proposed estimator: We improve the weighting for each sample pair to reduce the convergence error. We define the average of the proposed weights as $\hat{\Psi}_n$, then our estimation value as $\hat{n}$:

$$\hat{\Psi}_n = \frac{1}{|I|} \sum_{(k,l) \in I} \frac{d_{x_k}}{d_{x_l}^*}, \quad \hat{n} \triangleq \frac{\hat{\Psi}_n}{\Phi_n^{NC}}.$$

It holds the following lemma regarding the convergence values of the proposed estimator.

Lemma 4.3. *$\hat{n}$ converges to $\tilde{n} = n^* \frac{\sum_{v_i \in V^*} d_i^* d_i}{\sum_{v_i \in V^*} (d_i^*)^2}$.*

When there are no private nodes on G, the following proposition holds for each estimate and convergence value.

Proposition 4.4. *When there are no private nodes on G, the two estimation values, n^{NC} and $\hat{n}$, are equal, and the two convergence values, n^* and $\tilde{n}$, are equal to the true value, n.*

We clarify our convergence value, $\tilde{n}$, has the smaller expected error for any values of p than the previous convergence value, n^*. First, we show the following lemma:

Lemma 4.5. *For any node $v_i \in V^*$, we have*

$$E_{pri}[d_i^*] = (1 - p)d_i,$$
$$E_{pri}\left[(d_i^*)^2\right] = (1 - p)d_i[(1 - p)d_i + p].$$

Then, we approximate the expectation of $\tilde{n}$ regarding $\mathcal{L}_{pri}$ by deriving each expectation of the denominator and numerator of $\tilde{n}$.

Theorem 4.6. *Let $\alpha_p = \frac{(1-p)\sum_{v_i \in V} (d_i)^2}{\sum_{v_i \in V} d_i[(1-p)d_i + p]}$. Under the condition that all the public nodes belong to the largest public-cluster, we have*

$$E_{pri}[\tilde{n}] \approx \frac{E_{pri}[n^*] E_{pri}[\sum_{v_i \in V^*} d_i^* d_i]}{E_{pri}[\sum_{v_i \in V^*} (d_i^*)^2]} = \alpha_p n.$$

Theorem 4.6 implies that our convergence value is almost equal to the entire size because the coefficient α_p is empirically almost equal to 1 for various values of p in real social networks.

The following corollary claims that the proposed estimator improves an expected convergence error for any values of p.

Corollary 4.7. *Under the condition that all the public nodes belong to the largest public-cluster, we have*

$$\left| n - \frac{E_{pri}[n^*] E_{pri}[\sum_{v_i \in V^*} d_i^* d_i]}{E_{pri}[\sum_{v_i \in V^*} (d_i^*)^2]} \right| \leq \left| n - E_{pri}[n^*] \right|.$$

4.3 Average Degree Estimation

Existing estimator: The Smooth algorithm [6] takes a rough estimate of the average degree, c, as an input and returns a more accurate estimate. Herein, we let c to be 0 because Dasgupta et al. concluded that small values, 0 or 1, are desirable [6].

An estimate of the average degree obtained using the Smooth algorithm, d_{avg}^{Smooth}, is defined as follows:

$$d_{avg}^{Smooth} \triangleq \frac{r}{\sum_{k=1}^{r} \frac{1}{d_{x_k}^*}}.$$

It holds the following lemma derived from the previous study [6] which assumes a graph with no private nodes.

Lemma 4.8. *d_{avg}^{Smooth} converges to the average degree of the largest public-cluster, d_{avg}^*.*

Then, we approximate the expectation of d_{avg}^* regarding $\mathcal{L}_{pri}$.

Lemma 4.9. *Under the condition that all the public nodes belong to the largest public-cluster, we have*

$$E_{pri}[d_{avg}^*] \approx \frac{E_{pri}[D^*]}{E_{pri}[n^*]} = (1 - p)d_{avg}.$$

Lemma 4.9 implies that an expected relative error of the convergence value in the existing estimator is $1 - p$.

Proposed estimator: We improve the weighting to reduce a convergence error due to the value of p. We define our estimate as

$$\hat{d}_{avg} \triangleq \frac{r}{\sum_{k=1}^{r} \frac{1}{d_{x_k}}}.$$

The following lemma regarding our estimate holds:

Lemma 4.10. *$\hat{d}_{avg}$ converges to $\tilde{d}_{avg} = \frac{D^*}{\sum_{v_i \in V^*} \frac{d_i^*}{d_i}}$.*

The following proposition holds as well as Proposition 4.4.

Proposition 4.11. *When there are no private nodes on G, two estimates, d_{avg}^{Smooth} and $\hat{d}_{avg}$, are equal, and two convergence values, d_{avg}^* and $\tilde{d}_{avg}$, are equal to the true value, d_{avg}.*

The following theorem and corollary claim that the proposed estimator improves an expected error of the convergence value.

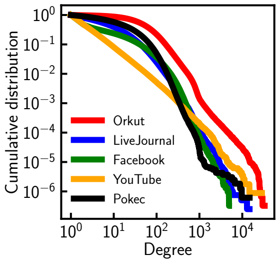


Figure 2: Cumulative degree distributions.

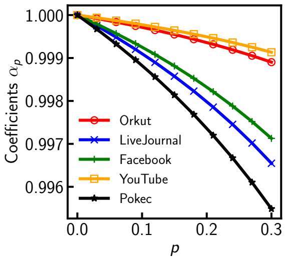


Figure 3: Coefficients α_p.

THEOREM 4.12. *Under the condition that all the public nodes belong to the largest public-cluster, we have*

$$E_{pri}[\tilde{d}_{avg}] \approx \frac{E_{pri}[D^*]}{E_{pri}\left[\sum_{v_i \in V^*} \frac{d_i^*}{d_i}\right]} = d_{avg}.$$

COROLLARY 4.13. *Under the condition that all the public nodes belong to the largest public-cluster, we have*

$$\left| d_{avg} - \frac{E_{pri}[D^*]}{E_{pri}\left[\sum_{v_i \in V^*} \frac{d_i^*}{d_i}\right]} \right| \leq \left| d_{avg} - \frac{E_{pri}[D^*]}{E_{pri}[n^*]} \right|.$$

4.4 Estimation in the Hidden Privacy Model

In the hidden privacy model, we calculate each estimate using the approximated public-degree, $\hat{d}^*_{x_k}$. Even in this model, Lemmas 4.1, 4.3, 4.8, and 4.10 hold because of Lemma 3.2.

5 EXPERIMENTS

We evaluate the proposed algorithms using real social network datasets. We aim to answer the following questions:

(1) Do the proposed estimators improve the estimation accuracy of the size and average degree for various probabilities of p? (Section 5.2)
(2) Do the proposed estimators perform acceptably in real-world datasets including real private users? (Section 5.3)
(3) How does the proposed public-degree calculation for the hidden privacy model affect the estimation accuracy and number of queries? (Section 5.4)
(4) Is the number of queries performed in the seed selection due to private nodes is small? (Section 5.5)

5.1 Experimental Setup

Datasets: We use datasets of known size and average degree, YouTube, Pokec, Orkut, Facebook, and LiveJournal. For these five datasets, we focus on undirected and connected graphs by the pre-processing: (1) removing the directions of edges if the original graphs are directed and then (2) deleting the nodes that are not contained in the largest connected component of the original graph. All the following experiments are unaffected because the above pre-processing is performed before setting privacy labels and any processing is not added to the graph after the privacy labels are set. The Pokec dataset contains all graph data of the Pokec network, including 552525 real private users (about 33.8%) and the real privacy labels. Table 1 lists the size and average degree of five datasets. Additionally, we use a dataset [15] of 1,016,275 samples of real public Facebook users via random walk during October 2010 to evaluate the actual performance of the proposed estimators. This dataset contains the ID, exact public-degree, and exact degree of each sampled public user.

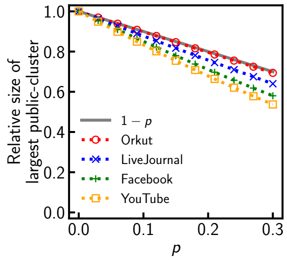


Figure 4: Relative size of the largest public-cluster.

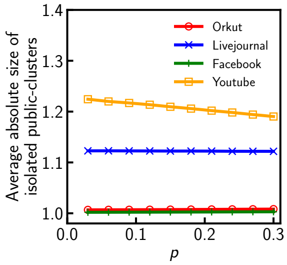


Figure 5: Average absolute size of isolated public-clusters.

Table 1: Datasets.

Network	n	d_{avg}	Privacy label setting
YouTube [14]	1,134,890	5.27	based on Assumption 2
Pokec [18]	1,632,803	27.32	real labels
Orkut [18]	3,072,441	76.28	based on Assumption 2
Facebook [24]	3,097,165	15.28	based on Assumption 2
LiveJournal [18]	3,997,962	17.35	based on Assumption 2

Accuracy measure: We apply the normalized root mean square error (NRMSE) to evaluate both the bias and variance of estimators [5, 11, 17, 28]. NRMSE is defined by $(E[(\frac{\tilde{x}}{x}-1)^2])^{\frac{1}{2}}$, where $\tilde{x}$ denotes the estimation value and x denotes the true value. We performed each experiment independently 1000 times to estimate the NRMSE.

Privacy label settings: For YouTube, Orkut, Facebook, and LiveJournal, we independently set each privacy label as private with probability p and otherwise public, according to Assumption 2. The probability p is from 0.0 to 0.30 in increments of 0.03 because there were actually tens of percentages of private users [4, 7, 25]. The set of privacy labels is independently given for each experiment. For Pokec, we apply the set of real privacy labels contained in the dataset. Table 1 lists how to set privacy labels in five datasets.

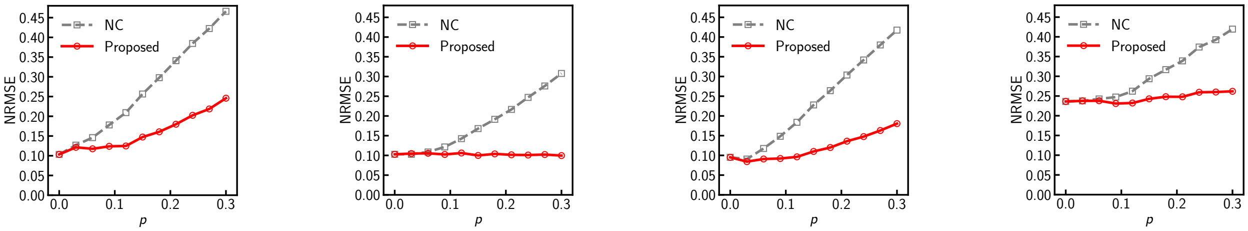
Seed selection: On YouTube, Orkut, Facebook, LiveJournal, and Pokec, the seed of a random walk is selected randomly from nodes on the largest public-cluster and independently for each experiment.

Sample size: The sample size, i.e., the length of a random walk, r, is 1% of the total number of nodes on YouTube, Orkut, Facebook, and LiveJournal. On Pokec, the sample size is varied from 0.5% to 5% of the total number of nodes in increments of 0.5%.

Threshold in size estimators: The threshold, m, in size estimators is 2.5% of the sample size, as set in a previous study [11].

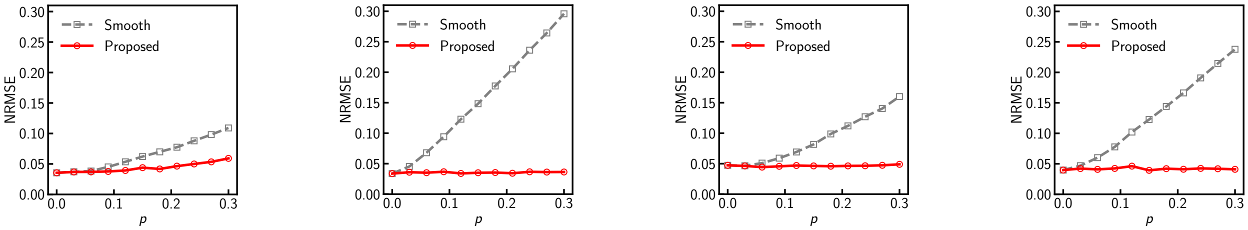
Degree distributions: Figure 2 shows the cumulative degree distributions of five datasets. We see that the degree distribution is biased to low degrees in all datasets.

Coefficients α_p: Figure 3 shows the coefficients α_p that is defined in Theorem 4.6 for various values of p in five datasets. We see that the coefficients α_p are almost equal to 1.0 for every value of p.



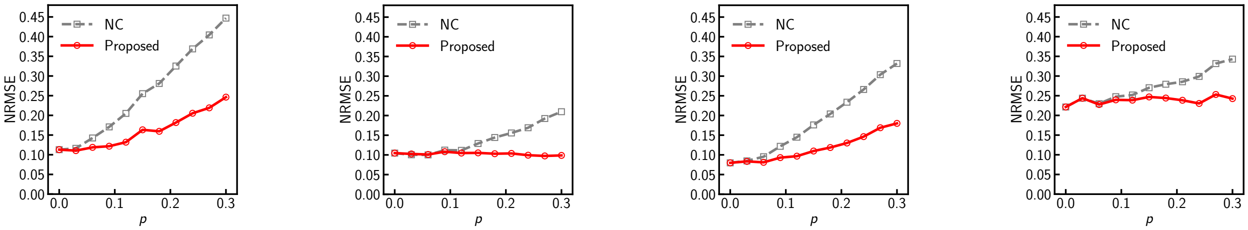
(a) YouTube (b) Orkut (c) Facebook (d) LiveJournal

Figure 6: NRMSEs of the size estimates (the ideal model with 1% sample size).



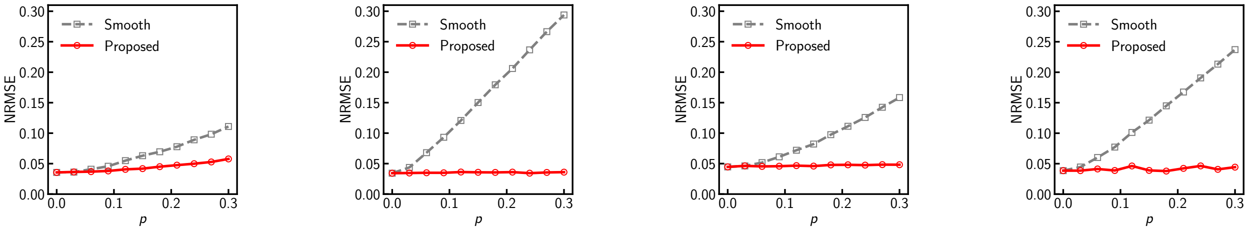
(a) YouTube (b) Orkut (c) Facebook (d) LiveJournal

Figure 7: NRMSEs of the average degree estimates (the ideal model with 1% sample size).



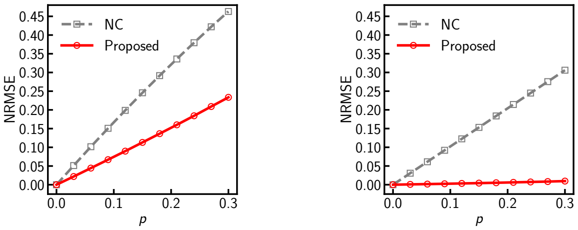
(a) YouTube (b) Orkut (c) Facebook (d) LiveJournal

Figure 8: NRMSEs of the size estimates (the hidden privacy model with 1% sample size).



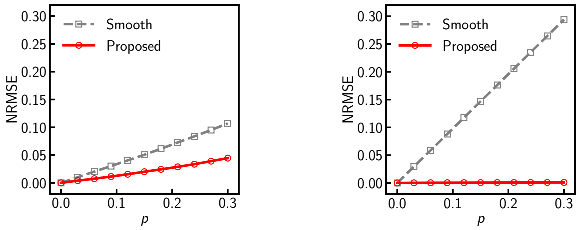
(a) YouTube (b) Orkut (c) Facebook (d) LiveJournal

Figure 9: NRMSEs of the average degree estimates (the hidden privacy model with 1% sample size).



(a) YouTube (b) Orkut

Figure 10: NRMSEs of size convergence values.



(a) YouTube (b) Orkut

Figure 11: NRMSEs of average degree convergence values.

Relative size of the largest public-cluster: Figure 4 shows the average relative size of the largest public-cluster, $\frac{n^*}{n}$, over independent 1000 experiments for various probabilities of p in four datasets. The upper limit is $1 - p$ (gray solid line). All the public nodes on Pokec belong to the largest public-cluster.

Average absolute size of isolated public-clusters: Figure 5 shows the average absolute size of isolated public-clusters (i.e., public-clusters other than the largest public-cluster) over independent 1000 experiments for various probabilities of p in four datasets. Pokec has no isolated public-clusters.

5.2 Estimation Accuracy for Percentage of Private Nodes

Figures 6 and 7 show the NRMSEs of each estimator for various probabilities of p with 1% sample size in the ideal model. The NRMSEs of both estimators are equal when $p = 0$ because of Propositions 4.4 and 4.11. The proposed estimators have lower NRMSEs for most probabilities of $p > 0$ than those of the existing estimators. For example, the proposed estimators improve the NRMSE approximately 88.1%, e.g., from 0.294 to 0.035, for $p = 0.3$ in Figure 7 (b). Figures 8 and 9 show that the proposed estimators improve the NRMSEs on all four datasets in the hidden privacy model as well.

The improvement of estimation errors results from that of expected errors of convergence values. Figures 10 and 11 show the NRMSEs of convergence values obtained from Lemmas 4.1, 4.3, 4.8 and 4.10 for various probabilities of p on YouTube and Orkut. The NRMSEs of both estimators are equal to 0 when $p = 0$ because of Propositions 4.4 and 4.11. The proposed estimators improve the NRMSEs of convergence values for any $p > 0$; this supports the claims of Corollaries 4.7 and 4.13. We have obtained similar results on Facebook and LiveJournal.

The errors of estimation and convergence values are affected by the relative size of the largest public-cluster because we sample the nodes only on the largest public-cluster. On Orkut where almost all the public nodes belong to the largest public-cluster (see Figure 4), the convergence values of proposed estimators are almost equal to true values for any probabilities of p (see Figures 10 and 11); this supports the claims of Theorems 4.6 and 4.12. Consequently, the NRMSEs of estimates on Orkut do not almost increase as the probabilities of p increases (see Figures 6, 7, 8, and 9). Conversely, on YouTube where there are the most public nodes that do not belong to the largest public-cluster among four datasets, the NRMSEs of the estimation and convergence values relatively increases.

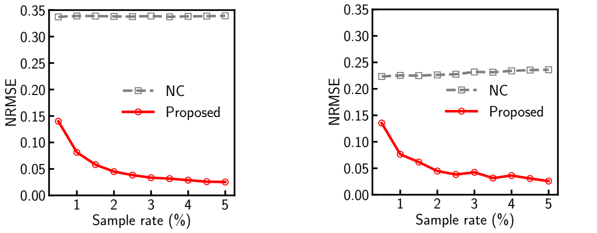
5.3 Estimation in Real-World Datasets

We evaluate the performance of the proposed estimators in two realistic datasets including real private users.

5.3.1 *Pokec.*

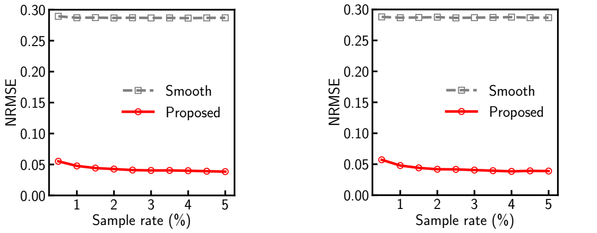
We use the Pokec dataset [18, 25] that contains all the graph data and real privacy labels of the Pokec network. There are 552525 private nodes (approximately 33.8%) on the Pokec graph. We evaluate errors of estimates and convergence values obtained by the existing and proposed estimators using this dataset.

Figures 12 and 13 show the NRMSEs of each estimator for various sample sizes. In both access models, the proposed estimators improve the NRMSEs for all sample sizes. The proposed size estimator particularly improves the NRMSE by approximately 92.6%, e.g., from 0.339 to 0.025, with a 5% sample size in Figure 12 (a).



(a) The ideal model (b) The hidden privacy model

Figure 12: NRMSEs of size estimates on Pokec.



(a) The ideal model (b) The hidden privacy model

Figure 13: NRMSEs of average degree estimates on Pokec.

Table 2: Convergence values (CV) and relative errors (RE) in Pokec (left column), and estimates obtained from samples of public Facebook Users (right column).

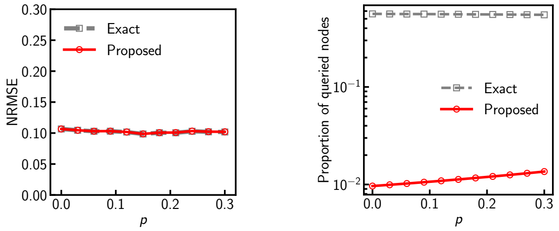
	Pokec		Real Facebook
Algorithm	CV	RE	Estimate
NC	1080278.0	0.338	480298540.0
Proposed	1646880.8	**0.009**	656874080.9
Smooth	19.49	0.287	102.07
Proposed	28.31	**0.036**	137.03

The improvement of estimation errors results from that of the errors of convergence values. The left column in Table 2 presents the convergence values obtained from Lemmas 4.1, 4.3, 4.8 and 4.10 and relative errors of each estimator. The proposed estimators improve the relative errors by 97.3%, e.g., from 0.338 to 0.009, for the size and 87.5%, e.g., from 0.287 to 0.036 for the average degree.

Moreover, the proposed estimators have convergence values that are almost equal to the true properties of the whole Pokec graph including private nodes. Even though Assumption 2 does not hold for the Pokec graph, we have obtained reasonable results as claimed in Theorems 4.6 and 4.12.

5.3.2 *Real public Facebook user samples.*

We use a dataset [15] of 1,016,275 real public Facebook user samples obtained by a random walk during October 2010. We can estimate the size and average degree of Facebook as of October 2010 using existing and proposed



(a) NRMSEs of the proposed size estimator

(b) Proportion of queried nodes

Figure 14: Effects of the proposed calculation for public-degrees on Orkut (the hidden privacy model with 1% sample size).

estimators, because this dataset contains the ID, exact public-degree, and exact degree of each sampled public user.

The right column in Table 2 shows estimates of each estimator. We set a threshold, m, in size estimators as 2.5% of the sample size, i.e., $m = 25407$. Although we cannot evaluate each error because the true size and average degree of Facebook as of October 2010 are unknown, we believe that estimates are reasonable, considering two findings in almost the same period.

First, Facebook reported that there were 500 million *active* users as of July 2010 [8]. This suggests that there were at least 500 million users including inactive users at that time. Notably, the estimated size in Table 2 includes both active and inactive users. Our estimate, i.e., 657 million, is greater than 500 million, and we speculate that the difference (approximately 157 million) includes inactive users.

Second, Catanese et al. obtained an unbiased estimate of the proportion of private users as 0.266 from uniform samples of Facebook users in August 2010 [4]. From Lemmas 4.2 and 4.9 and Theorems 4.6 and 4.12, we can intuitively estimate the value of p, which is almost equal to the proportion of private users, from the respective estimates as $\hat{p}_n \triangleq 1 - \frac{n^{NC}}{\hat{n}}$ and $\hat{p}_{avg} \triangleq 1 - \frac{d^{Smooth}_{avg}}{\hat{d}_{avg}}$. Here, $\hat{p}_n$ and $\hat{p}_{avg}$ are the estimates of p obtained by the existing and proposed estimators for the size and average degree, respectively. From Table 2, we obtain $\hat{p}_n$ and $\hat{p}_{avg}$ as 0.269 and 0.255, respectively, which are considerably close to the ground truth value of 0.266.

5.4 Effectiveness of the Proposed Public-Degree Calculation

We evaluate the proposed public-degree calculation for the hidden privacy model. The proposed size estimator requires the public-degree of each sample (see Section 4.2). Thus, we compare the size estimation accuracy and proportion of queried nodes of the exact and proposed methods. The exact method queries all neighbors of each sample to use the exact public-degree in weighting.

Figure 14 shows the experimental result with a 1% sample size on Orkut. The proposed approximation provides almost the same estimation accuracy as the exact method (see Figure 14 (a)). The proposed method queries approximately 1% nodes; however, the exact method queries over 50% nodes which are much greater than 1% sample size (see Figure 14 (b)). The proposed method reduces the proportion of queried nodes by tens of percent with almost the same accuracy on other datasets as well.

5.5 Seed Selection from the Largest Public-Cluster

In practical scenarios, we may require additional queries caused by restarting a random walk from another seed in the two cases: (1) when a selected seed is a private node and (2) when a selected public seed is on an isolated public-cluster.

We believe that the number of queries performed in each case is sufficiently small. For the case of (1), the proportion of private nodes is empirically smaller than that of public nodes, e.g., 27% on Facebook [4] and 34% on Pokec [25]. Additionally, one query is enough to check the privacy label of a node. For the case of (2), we believe that the number of queries is small under Assumption 2, because of the following two natures of graphs with degree distributions biased to low degrees [2]. First, most public nodes belong to the largest public-cluster. In our experiments, even if $p = 0.3$, 99.1% , e.g., $\frac{0.694}{0.7}$ in Orkut, 91.4%, e.g., $\frac{0.640}{0.7}$ in LiveJournal, 82.8%, e.g., $\frac{0.580}{0.7}$ in Facebook and 76.7%, e.g., $\frac{0.537}{0.7}$ in YouTube of public nodes belong to the largest public-cluster on average (see Figure 4). Therefore, a selected public seed will belong to the largest public-cluster with high probability. Second, the size of the isolated public-clusters is considerably smaller than that of the largest public-cluster. In our experiments, the average absolute size is only approximately one for every probability of p (see Figure 5). Therefore, few queries are performed on isolated public-clusters.

We believe that two real-world datasets support Assumption 3. There is only the largest public-cluster on Pokec; the case of (2) cannot occur. The samples of real public Facebook users provide an estimated size of 657 million and contain one million unique samples; we believe that the samples were obtained from the largest public-cluster of Facebook as of October 2010.

6 RELATED WORK

Crawling techniques that traverse neighbors are effective to sample graph data in OSNs where neighbor data is obtained through queries. The breadth-first search is particularly a basic sampling technique applied in early studies on analyzing the graph structure of OSNs [1, 9, 10, 20, 29]. However, Gjoka et al., through the case study of crawling in Facebook, demonstrated that the breadth-first search with a small sample size leads to a significant sampling bias that is difficult to correct [9, 10].

Gjoka et al. designed a practical framework to obtain unbiased estimates of graph properties via a random walk-based sampling in social networks [9, 10]. Gjoka et al. clarified that the re-weighted random walk scheme, where each sample via random walk is weighted to correct the sampling bias, is effective to obtain unbiased estimates. In this study, we have made the framework of Gjoka et al. more practical by addressing the effects of private nodes on the re-weighted random walk scheme for the first time. Specifically, we have designed a random walk-based sampling algorithm considering private nodes and proposed the weighting methods to reduce both sampling bias and estimation errors due to private nodes.

The estimation algorithms based on the re-weighted random walk scheme have been studied for several properties [5, 6, 11, 12, 22, 28]. The targeted properties widely range from those of the graph such as the size and the average degree to those of the nodes like centralities. We have designed the algorithms to accurately estimate the size and average degree of the whole graph including private nodes. This study has focused on estimators for only the size and average degree; however, there is no particular reason to be limited to these. The goal of weighting methods to reduce both sampling bias and errors due to private nodes is shared in designing all of the random walk-based estimators for social networks. Then, the methodology presented in our study will help these designs.

Ye et al. investigated the effects of private nodes on general crawling methods and empirically argued that private nodes cause only a small reduction rate of the number of nodes and edges that can be sampled by crawling, under Assumptions 1 and 2 [30]. However, Ye et al. did not support the validity of assumptions. In this study, we have focused on the effects of private nodes on random walk-based algorithms. Moreover, we have validated the three assumptions through experiments using real-world datasets.

Kossinets experimentally discussed the effect of missing the nodes and edges on the properties of social networks [13]. Kossinets reported that as the proportion of randomly deleted nodes increases, errors of the size and average degree between the original graph and the remaining largest connected component increase. We have theoretically analyzed these errors and designed algorithms to improve them under certain assumptions and access models. The important differences from the study of Kossinets are as follows: (i) we can find most private nodes in neighbors of public nodes on social networks under Assumption 1 and (ii) we can reduce the errors due to private nodes by modifying the weighs for each sample obtained from the largest public-cluster based on Assumption 2.

7 CONCLUSIONS

We have designed re-weighted random walk algorithms considering private nodes for the first time. The previous studies have not addressed the effects of private nodes on re-weighted random walk algorithms because private nodes prevent us from performing a simple random walk on a social graph. We have broken through this situation by making three assumptions. In particular, Assumption 2 which is seemingly too simple has brought surprisingly reasonable results in experiments using real-world datasets. Our study has opened the door to accurately estimate the whole graph topology of social networks including private nodes.

There are mainly two future works. First, we plan to strengthen the validity of three assumptions. To further verify Assumptions 1 and 2, we should additionally test the proposed estimators in real social networks including private nodes. To support Assumption 3 more, we will improve the sampling algorithm considering the case that the seed is on an isolated public-cluster. Second, we plan to design estimators considering private nodes for other properties, such as clustering coefficients [11] and motifs (or graphlets) [5, 28].

ACKNOWLEDGMENTS

We would like to thank Dr. Minas Gjoka for kind replies to our questions. We would like to thank Prof. Naoto Miyoshi for helpful comments. This work was supported by New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. 2007. Analysis of Topological Characteristics of Huge Online Social Networking Services. In *WWW*. 835–844.

[2] Réka Albert, Hawoong Jeong, and Albert-László Barabási. 2000. Error and attack tolerance of complex networks. *Nature* 406 (2000), 378–382.

[3] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. 2012. Four Degrees of Separation. In *WebSci*. 33–42.

[4] Salvatore A Catanese, Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. 2011. Crawling Facebook for Social Network Analysis Purposes. In *WIMS*. 1–8.

[5] Xiaowei Chen, Yongkun Li, Pinghui Wang, and John Lui. 2016. A General Framework for Estimating Graphlet Statistics via Random Walk. *PVLDB* 10, 3 (2016), 253–264.

[6] Anirban Dasgupta, Ravi Kumar, and Tamas Sarlos. 2014. On Estimating the Average Degree. In *WWW*. 795–806.

[7] Ratan Dey, Zubin Jelveh, and Keith Ross. 2012. Facebook Users Have Become Much More Private: A Large-Scale Study. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. 346–352.

[8] Facebook. 2010. Facebook 500 Million Stories. https://www.facebook.com/notes/facebook/500-million-stories/409753352130/.

[9] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. 2010. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In *INFOCOM*. 1–9.

[10] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. 2011. Practical Recommendations on Crawling Online Social Networks. *IEEE Journal on Selected Areas in Communications* 29, 9 (2011), 1872–1892.

[11] Stephen J Hardiman and Liran Katzir. 2013. Estimating clustering coefficients and size of social networks via random walk. In *WWW*. 539–550.

[12] Liran Katzir, Edo Liberty, and Oren Somekh. 2011. Estimating sizes of social networks via biased sampling. In *WWW*. 597–606.

[13] Gueorgi Kossinets. 2006. Effects of missing data in social networks. *Social networks* 28, 3 (2006), 247–268.

[14] Jérôme Kunegis. 2013. KONECT – The Koblenz Network Collection. In *WWW*. 1343–1350.

[15] Maciej Kurant, Minas Gjoka, Carter T Butts, and Athina Markopoulou. 2011. Walking on a Graph with a Magnifying Glass: Stratified Sampling via Weighted Random Walks. In *SIGMETRICS*. 281–292.

[16] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a Social Network or a News Media?. In *WWW*. 591–600.

[17] Chul-Ho Lee, Xin Xu, and Do Young Eun. 2012. Beyond random walk and metropolis-hastings samplers: why you should not backtrack for unbiased graph sampling. In *SIGMETRICS*. 319–330.

[18] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[19] David A Levin and Yuval Peres. 2017. *Markov Chains and Mixing Times*. Vol. 107. American Mathematical Soc.

[20] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *IMC*. 29–42.

[21] Seth A Myers, Aneesh Sharma, Pankaj Gupta, and Jimmy Lin. 2014. Information Network or Social Network? The Structure of the Twitter Follow Graph. In *WWW*. 493–498.

[22] Kazuki Nakajima and Kazuyuki Shudo. 2020. Estimating High Betweenness Centrality Nodes via Random walk in Social Networks. *Journal of Information Processing (JIP)* 28 (2020). (to appear).

[23] Bruno Ribeiro and Don Towsley. 2010. Estimating and Sampling Graphs with Multidimensional Random Walks. In *IMC*. 390–403.

[24] Ryan A. Rossi and Nesreen K. Ahmed. 2015. The Network Data Repository with Interactive Graph Analytics and Visualization. In *AAAI*. 4292–4293.

[25] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*.

[26] Twitter. 2020. Twitter API GET followers/ids. https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-followers-ids.html.

[27] Twitter. 2020. Twitter API GET friends/ids. https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-friends-ids.html.

[28] Pinghui Wang, John Lui, Bruno Ribeiro, Don Towsley, Junzhou Zhao, and Xiaohong Guan. 2014. Efficiently Estimating Motif Statistics of Large Networks. *TKDD* 9, 2 (2014).

[29] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna PN Puttaswamy, and Ben Y Zhao. 2009. User interactions in social networks and their implications. In *EuroSys*. 205–218.

[30] Shaozhi Ye, Juan Lang, and Felix Wu. 2010. Crawling Online Social Graphs. In *APWEB*. 236–242.

In the supplement, we show the pseudo code and proofs that could not be included in the main manuscript due to space limitations and the information for reproducing the experimental results.

A PSEUDO-CODE

Algorithm 1 Random walk in the hidden privacy model.

```
Require: seed $v_{x_1} \in C^*$, sample size $r$
Ensure: sampling list $R$
1:  $R \leftarrow$ empty list
2:  for $k = 1$ to $r$ do
3:      $d_{x_k} \leftarrow$ the degree of $v_{x_k}$
4:      $\hat{d}^*_{x_k} \leftarrow 0$
5:      append $(x_k, d_{x_k}, \hat{d}^*_{x_k}) \rightarrow R$
6:      if $v_{x_k}$ has been visited for the first time then
7:          $a_{x_k} \leftarrow 0$
8:          $b_{x_k} \leftarrow 0$
9:      end if
10:     flag $\leftarrow$ False
11:     while flag is False do
12:         $v_{x_{k+1}} \leftarrow$ a neighbor of $v_{x_k}$ chosen randomly
13:         $b_{x_k} \leftarrow b_{x_k} + 1$
14:         if $v_{x_{k+1}}$ is a public node then
15:             $a_{x_k} \leftarrow a_{x_k} + 1$
16:             flag $\leftarrow$ True
17:         end if
18:     end while
19: end for
20: for $k = 1$ to $r$ do
21:     $\hat{d}^*_{x_k} \leftarrow d_{x_k} \frac{a_{x_k}}{b_{x_k}}$
22: end for
23: return $R$
```

Algorithm 1 describes the designed random walk-based sampling using the proposed public-degree calculation in the hidden privacy model. The process in lines from 11 to 18 will be successfully performed, because the k-th sample, v_{x_k}, has at least one public neighbor $v_{x_{k-1}}$ for $2 \le k \le r$ and v_{x_1} is on the largest public-cluster. Additionally, it follows that $b_{x_k} > 0$ for each sample v_{x_k}, because a neighbor selection is performed at least once for each sample.

B PROOFS

We show proofs of Lemmas 3.1, 3.2, 3.3, 4.1, 4.2, 4.3, 4.5, 4.8, 4.9, 4.10, Proposition 4.4, Theorems 4.6, 4.12, and Corollary 4.7.

B.1 Proof of Lemma 3.1

Proof. First, it holds $Pr[x_r = i] = 0$ for each node $v_i \in V \backslash V^*$, because we cannot reach nodes not in V^*. Then, for each node $v_i \in V^*$, we show that $Pr[x_r = i]$ converges to $\frac{d^*_i}{D^*}$ after many steps. The designed random walk started from the seed on C^* has the transition probability matrix $\boldsymbol{P} = \{P_{i,j}\}_{v_i, v_j \in V^*}$, where $P_{i,j}$ is $\frac{1}{d^*_i}$, if $(v_i, v_j) \in E^*$, otherwise, 0. $\boldsymbol{P}$ is ergodic because C^* is a connected graph, and hence, the stationary distribution uniquely exists because of the Theorem 2.1. $\{p_i\}_{v_i \in V^*}$ satisfies the definition of the stationary distribution. The probability $Pr[x_r = i]$ converges to the corresponding stationary distribution after many steps. □

B.2 Proof of Lemma 3.2

Proof. Let $\hat{X}_k(l)$ denote a random variable which returns 1 if a public neighbor of v_{x_k} is selected at the l-th trial, otherwise 0, where $1 \le l \le b_{x_k}$ and it holds $\sum_{l=1}^{b_{x_k}} \hat{X}_k(l) = a_{x_k}$. It holds $Pr[\hat{X}_k(l) = 1] = \frac{d^*_{x_k}}{d_{x_k}}$ because $\hat{X}_k(l)$ follows a Bernoulli distribution for each l. Then, it holds $E[\hat{d}^*_{x_k}] = d_{x_k} \frac{d^*_{x_k}}{d_{x_k}} = d^*_{x_k}$. Because $\{\hat{X}_k(l)\}_{l=1}^{b_{x_k}}$ is a process of independent and identically distributed trials, $\hat{d}^*_{x_k}$ converges to its expectation after many steps of the designed random walk because of the law of large numbers. □

B.3 Proof of Lemma 3.3

Proof. It holds $Q(k) = d_{x_k}$, because the exact method queries all neighbors of each sample. Therefore, we have

$$E[Q] = E[Q(k)] = \sum_{v_i \in V^*} \frac{d^*_i}{D^*} E[Q(k) | x_k = i] = \frac{\sum_{v_i \in V^*} d^*_i d_i}{D^*}.$$

The first and second equations hold because of the linearity of expectation and the law of total expectation.

$\hat{Q}(k)$ follows the geometric distribution with success probability $\frac{d^*_{x_k}}{d_{x_k}}$, because the proposed method randomly queries neighbors of v_{x_k} until the public neighbor is firstly selected. Thus, it holds $E[\hat{Q}(k)] = \frac{d_{x_k}}{d^*_{x_k}}$. Then, we have

$$E[\hat{Q}] = E[\hat{Q}(k)] = \sum_{v_i \in V^*} \frac{d^*_i}{D^*} \frac{d_i}{d^*_i} = \frac{\sum_{v_i \in V^*} d_i}{D^*}.$$

Lemma 3.3 holds because of the above equations. □

B.4 Proof of Lemma 4.1

Proof. First, we calculate the expectation of Φ_n^{NC}:

$$E\left[\Phi_n^{NC}\right] = E[\phi_{k,l}] = \sum_{v_i \in V^*} \left(\frac{d^*_i}{D^*}\right)^2.$$

The first equation holds because of the linearity of expectation. The second equation holds because v_{x_k} and v_{x_l} are independently sampled with the probability $\frac{d^*_i}{D^*}$. Then, we obtain

$$E\left[\Psi_n^{NC}\right] = E\left[\frac{d^*_{x_k}}{d^*_{x_l}}\right] = \sum_{v_i \in V^*} \sum_{v_j \in V^*} \frac{d^*_i}{d^*_j} \frac{d^*_j}{D^*} \frac{d^*_i}{D^*} = n^* \sum_{v_i \in V^*} \left(\frac{d^*_i}{D^*}\right)^2.$$

We conclude that n^{NC} converges to n^*, because Φ_n^{NC} and Ψ_n^{NC} intuitively converge to their respective expectations. □

B.5 Proof of Lemma 4.2

Proof. We define the random variable $X_n(i) = \mathbf{1}_{V^*}(v_i)$ for each node $v_i \in V$. It holds $n^* = \sum_{v_i \in V} X_n(i)$. The expected value of n^* regarding $\mathcal{L}_{pri}$ is given by:

$$E_{pri}[n^*] = \sum_{v_i \in V} E_{pri}[X_n(i)] = \sum_{v_i \in V} Pr[v_i \in V^*] = (1-p)n.$$

The first and second equations hold because of the linearity of the expected value and the law of total expectation. □

B.6 Proof of Lemma 4.3

Proof. As with the proof of the Lemma 4.1, we have

$$E\left[\hat{\Psi}_n\right] = E\left[\frac{d_{x_k}}{d^*_{x_l}}\right] = n^* \sum_{v_i \in V^*} \frac{d^*_i d_i}{(D^*)^2}.$$

Because Φ_n^{NC} and $\hat{\Psi}_n$ intuitively converge to the expected values respectively, $\hat{n}$ converges to $\tilde{n}$. □

B.7 Proof of Proposition 4.4

Proof. When there are no private nodes on G, we have $V^* = V$ and $d^*_i = d_i$ for each node $v_i \in V^*$, because C^* is equivalent to G. Thus, it holds $n^{NC} = \hat{n}$ from the definitions. It holds $n^* = \tilde{n} = n$ because of Lemmas 4.1 and 4.3. □

B.8 Proof of Lemma 4.5

Proof. d^*_i follows the binomial distribution with parameters d_i and $1-p$ regarding $\mathcal{L}_{pri}$, because each neighbor of v_i independently becomes a public node with probability $1 - p$. □

B.9 Proof of Theorem 4.6

Proof. We define the random variables $\hat{X}_n(i) = d^*_i d_i \mathbf{1}_{V^*}(v_i)$ and $\hat{Y}_n(i) = (d^*_i)^2 \mathbf{1}_{V^*}(v_i)$ for each node $v_i \in V$. Also, let $\hat{X} = \sum_{v_i \in V^*} d^*_i d_i$ and $\hat{Y} = \sum_{v_i \in V^*} (d^*_i)^2$. It holds $\hat{X} = \sum_{v_i \in V} \hat{X}_n(i)$ and $\hat{Y} = \sum_{v_i \in V} \hat{Y}_n(i)$. We obtain the expectation of $\hat{X}$ regarding $\mathcal{L}_{pri}$:

$$E_{pri}[\hat{X}] = \sum_{v_i \in V} Pr[v_i \in V^*] E_{pri}[d^*_i d_i] = (1-p)^2 \sum_{v_i \in V} (d_i)^2.$$

The second equation hold because of Lemma 4.5. We note that the degree d_i is a constant regarding $\mathcal{L}_{pri}$. Similarly, the expectation of $\hat{Y}$ regarding $\mathcal{L}_{pri}$ can be obtained as follows:

$$\begin{aligned} E_{pri}[\hat{Y}] &= \sum_{v_i \in V} Pr[v_i \in V^*] E_{pri}[(d^*_i)^2] \\ &= (1-p)^2 \sum_{v_i \in V} d_i[(1-p)d_i + p]. \end{aligned}$$

Theorem 4.6 holds because of above equations and Lemma 4.2. □

B.10 Proof of Corollary 4.7

Proof. From Lemma 4.2 and Theorem 4.6, it is sufficient to show that $|1 - \alpha_p| \leq p$. First, we show that $1 - \alpha_p \geq 0$:

$$1 - \alpha_p = \frac{p \sum_{v_i \in V} d_i}{\sum_{v_i \in V} d_i[(1-p)d_i + p]} \geq 0.$$

The inequality holds because it holds $d_i \geq 1$ for each node $v_i \in V^*$ and $0 \leq p \leq 1$. It holds equality when $p = 0$. Therefore, it is enough to show that $1 - \alpha_p \leq p$. We have

$$1 - \alpha_p - p = -(1-p)p \frac{\sum_{v_i \in V} [d_i(d_i - 1)]}{\sum_{v_i \in V} d_i[(1-p)d_i + p]} \leq 0.$$

The inequality holds since it holds $d_i \geq 1$ for each node $v_i \in V^*$ and $0 \leq p \leq 1$. □

B.11 Proof of Lemma 4.8

Proof. Let Φ_{avg}^{Smooth} be the weighted average of $\frac{1}{d^*_{x_k}}$ for samples. We calculate the expectation of Φ_{avg}^{Smooth} as follows:

$$E\left[\Phi_{avg}^{Smooth}\right] = E\left[\frac{1}{d^*_{x_k}}\right] = \sum_{v_i \in V^*} \frac{d^*_i}{D^*} \frac{1}{d^*_i} = \frac{1}{d^*_{avg}}.$$

Because Φ_{avg}^{Smooth} converges to the expected value after many steps because of Theorem 2.2, d_{avg}^{Smooth} converges to d^*_{avg}. □

B.12 Proof of Lemma 4.9

Proof. We define the random variable $X_{avg}(i) = d^*_i \mathbf{1}_{V^*}(v_i)$ for each node $v_i \in V$. It holds $D^* = \sum_{v_i \in V} X_{avg}(i)$. The expected value regarding $\mathcal{L}_{pri}$ of D^* can be derived:

$$E_{pri}[D^*] = \sum_{v_i \in V} Pr[v_i \in V^*] E_{pri}[d^*_i] = (1-p)^2 D.$$

Consequently, it follows that Lemma 4.9 holds because of the above equation and Lemma 4.2. □

B.13 Proof of Lemma 4.10

Proof. Let $\hat{\Phi}_{avg} = \frac{1}{r} \sum_{k=1}^{r} \frac{1}{d_{x_k}}$ be the weighted average of $\frac{1}{d_{x_k}}$ for samples. Then, we have:

$$E[\hat{\Phi}_{avg}] = E\left[\frac{1}{d_{x_k}}\right] = \sum_{v_i \in V^*} \frac{d^*_i}{D^*} \frac{1}{d_i} = \frac{1}{\tilde{d}_{avg}}.$$

Since $\hat{\Phi}_{avg}$ converges to the expected value after many steps because of Theorem 2.2, $\hat{d}_{avg}$ converges to $\tilde{d}_{avg}$. □

B.14 Proof of Theorem 4.12

Proof. We define the random variable $\hat{X}_{avg}(i) = \frac{d^*_i}{d_i} \mathbf{1}_{V^*}(v_i)$ for each node $v_i \in V$. Also, let $\hat{X}_{avg} = \sum_{v_i \in V^*} \frac{d^*_i}{d_i}$. It holds $\hat{X}_{avg} = \sum_{v_i \in V} \hat{X}_{avg}(i)$. We obtain the expectation of $\hat{X}_{avg}$ regarding $\mathcal{L}_{pri}$:

$$E_{pri}[\hat{X}_{avg}] = \sum_{v_i \in V} Pr[v_i \in V^*] E_{pri}\left[\frac{d^*_i}{d_i}\right] = (1-p)^2 n.$$

Using the fact that $E_{pri}[D^*] = (1-p)^2 D$ and the above equation, we have that the Theorem 4.12 holds. □

C ADDITIONAL INFORMATION FOR REPRODUCING EXPERIMENTAL RESULTS

All the original datasets are publicly available as follows:
YouTube: http://konect.uni-koblenz.de/networks/com-youtube
Pokec: http://snap.stanford.edu/data/soc-Pokec.html
Orkut: https://snap.stanford.edu/data/com-Orkut.html
Facebook: http://networkrepository.com/socfb-A-anon.php
LiveJournal: https://snap.stanford.edu/data/com-LiveJournal.html
Real public Facebook user samples: http://odysseas.calit2.uci.edu/doku.php/public:osn_datasets#facebook_weighted_random_walks

The datasets and source code that were used to generate the results in this paper are available at https://github.com/kazuibasou/KDD2020.