De-anonymization of Social Networks: the Power of Collectiveness

Jiapeng Zhang, Luoyi Fu, Xinbing Wang and Songwu Lu
{zhangjape, yiluofu, xwang8}@sjtu.edu.cn, slu@cs.ucla.edu

***Abstract*—The interaction among users in different social networks raises deep concern on user privacy, as it may facilitate the assailants to identify user identities by matching the anonymized networks with a correlated sanitized one. Prior arts regarding such de-anonymization problem can be primarily divided into a seeded case or a seedless one, depending on whether or not there are a subset of pre-identified nodes. The seedless case is much more complicated since the adjacency matrix representation of one-hop user relations delivers limited structural information.**

To address this issue, we, for the first time, integrate the multi-hop neighborhood relationships, which exhibit more structural commonness between the anonymized and the sanitized networks, into seedless de-anonymization process. Our aim is to sufficiently leverage these multi-hop neighbors of all nodes and minimize the total disagreements of these multi-hop adjacency matrices, which we call *collective adjacency disagreements* (CADs), between two networks of different sizes. Theoretically, we demonstrate that CAD enlarges the difference between wrongly matched node pairs and correctly matched pairs, whereby two networks can be correctly matched with high probability even when the network density is below $\log n$. Algorithmically, we adopt the conditional gradient descending method on a collective-form objective, which can efficiently find the minimal CADs for networks with broad degree distributions. Experiments on both synthetic and real-world networks return desirable de-anonymization accuracies thanks to the rich structural information manifested by such collectiveness, since most nodes can be correctly matched with their correspondences, especially in sparse networks where merely utilizing adjacency relations might fail to work.

I. Introduction

A wave of interaction between different social networks brings potential risk of privacy disclosure with the advancement of nowadays de-anonymization techniques [1], [2]. Although some social networks remove personal identifiers when they publish their data to third-parties, there could still be some other sanitized social networks whose data are available to the public. Therefore, the adversaries may re-identify these data by mapping the network structure to the sanitized ones.

Such process of unveiling users' identities by leveraging information from other domains is defined as social network de-anonymization, which is initiated by Narayanan and Shmatikov [3]. The de-anonymization problem has received conscionable attention so far, which is usually formulated based on a common paradigm that will also be adopted in our work. In this paradigm, an underlying network characterizes the potential relationships between users, while the adversaries can observe two networks, i.e., an anonymized one and a sanitized one, whose node sets and edges are independently sampled from the underlying network. Since the two observed networks are different but correlated, the aim of de-anonymization is to discover the correct matching between the two observed networks, which may contain partially the same users and user relationships, with the network structure as the only side information available to the adversaries.

Known as being equivalent to the NP-hard quadratic assignment problem [4], the de-anonymization problem can be primarily classified into the seeded or seedless case, depending on whether there exist some pre-identified nodes for inference. Abundant researches [5], [6] have been devoted to the seeded case, where node can be incrementally matched in terms of their adjacency relations with the aid of a small set of seeds. In contrast, the seedless case turns out to be more challenging [7], due either to the insufficient auxiliary information for the direct matching via a comparison between the adjacency matrices or to the high complexity for the indirect matching via evaluating the nodal or edge similarities.

In this paper, we probe into the seedless de-anonymization problem, where our particular concern is the tradeoff between the incomplete utilization of the structural information and the high computational complexity. On the one hand, if we depend merely on the adjacency relationships [8] in trade for efficiency, the local topologies of nodes may not bring desirable matching accuracy in solving this problem, as existing researches [5], [6], [9] on different network models usually set demands of a large mean degree $\bar{k} = \Omega(\log n)$. On the other hand, to explore richer structural commonalities between two networks, the similarity matrix or Kronecker product [10] is often referred, which, however, may bring a high computational complexity of $\Omega(n^4)$. Therefore, the question naturally arises: is it possible to solve the de-anonymization problem both effectively and efficiently?

To answer this question, we note that in social networks, there is an interesting phenomenon named friendship paradox [11], where, on average, the number of friends of our random friend is always greater than or equal to the number of ourselves' friends. Put differently, the neighbors of a user may expose more information than the user himself/herself. This phenomenon inspires us to collect one's friends, the friends of his/her friends and so forth to assist with user identification in the de-anonymization problem of interest. Particularly, in the common paradigm mentioned earlier, we collect the multi-hop neighborhood relations between different node pairs in both the sanitized and anonymized networks, and aim to match the users in the two networks by minimizing the total number of the mismatched relations within l hops. This number is called *collective adjacency disagreement* (CAD) at level l. As

will be illustrated later, CAD is capable of revealing richer structural information, thus leading to an enhancement of matching accuracy (a dramatic increase from 0 to 1 in some cases) without incurring extra computational complexity.

With the collectiveness taken into consideration, the purposes of this paper are two-folded: 1) we aim to improve the matching performance by virtue of our newly introduced collective adjacency, which unveils more side structural information that can assist in de-anonymization process; 2) we attempt to realize the de-anonymization by efficiently minimizing CAD, which is approximated by a convex problem that can be solved with relatively low computational complexity. Thereafter, we can unfold our contributions as follows:

1. We formalize the social network de-anonymization problem in the context of multi-hop adjacency relationship, and further generate the collective-form de-anonymization problem which aims to minimize the total differences between multi-hop adjacency matrices of the two observed networks called collective adjacency disagreement (CAD), which is better behaved for networks with broad degree distributions.

2. We theoretically derive the conditions for networks with arbitrary edge existence distributions to be correctly matched by ensuring that the correct matching possesses the minimal CAD, from which we also indicate an upper bound of the number of wrong matches that have lower adjacency disagreements than the correct matching. Specifically, networks with mean degree $\Omega((\log n)^{1/l})$ can be successfully de-anonymized with high probability when assisted by the CAD at level l.

3. In view of the NP-hardness to find the correct matching, we incorporate the collectiveness into the Fast Approximating Quadratic programming [8] and propose a Collective De-anonymizaion Algorithm (CDA). This algorithm is further refined by our RCDA algorithm in terms of both efficiency and effectiveness via the combination of an $O(n^2)$-time approximation method and a pre-sorting of nodes based on their degrees. As validated empirically, RCDA incurs less runtime while retaining comparable or even better accuracy than CDA.

Extensive experiments are performed on different kinds of networks, including synthetic networks such as Erdős-Rényi networks [12], Scale-Free networks [13], and real-world networks with unknown distributions. Empirical results suggest that smaller CAD usually brings higher matching accuracy, while the adjacency disagreement is poorly relevant with the accuracy when networks cannot be fully correctly matched.

II. Related Works

Social network de-anonymization is of rich and evolving concern in recent decades. Originally officially formulated by Narayanan and Shmatikov [3], this problem has been concretized in both theoretical and experimental aspects supported with a large amount of literature. These arts chiefly probe into two distinct categories of de-anonymization techniques, namely seeded and seedless attacks from the adversaries.

In the seeded case, a small set of nodes will be pre-identified [3]. On this basis, a frequently utilized method is bootstrap percolation, which can successively de-anonymize the neighbors of identified nodes with a handful of seeds under both Erdős-Rényi network model [5] and Scale-Free network model [14]. A similar method is also proposed in [15] to de-anonymize the nodes based on their local neighborhood relationships. Further, Nilizadeh et al. [16] improve the seeded de-anonymization scheme at a community level, whereby the users within a same de-anonymized community can be further de-anonymized with existing methods.

In contrast, the seedless case puts forward higher requirements to the adversaries since nodes can hardly be incrementally de-anonymized. Pedarsani and Grossglauer [17] mathematically build a random graph model, where two observed networks are obtained by independently sampling on edges of an underlying network, and derive the condition for correct matching in the seedless de-anonymization problem. Kazemi et al. [18] facilitate this model with node sampling, and reveal that networks can be correctly de-anonymized when the mean degrees are $\Omega(\log n)$. Practically, algorithms such as [19] by minimizing the adjacency disagreements are proposed to realize 'perfect' de-anonymization. Community-based seedless attacks are also delved into to achieve optimal performance under an MAP [20] or MMSE [21] estimator.

However, the literature above mainly takes advantage of the nearest neighborhood relations among nodes. In social networks, it is clarified that a perspective of two-hop neighbors "makes it easier for the contagion to prevail" [22]. As stated by Morone and Makse [23], a node could be influential to not only its directly connected neighbors, but also multi-hop neighbors. Analogously, to de-anonymize the nodes from the social network, the multi-hop neighbors for a node can also be helpful in identifying its special property. In [24], for instance, it is shown that the performance improves significantly in seeded graph matching with l-hop local neighborhoods. In this work, we combine the efficiency of FAQ algorithm and the rich features of the collectiveness, aiming to solve the seedless de-anonymization problem both efficiently and effectively.

III. Network Model and Problem Formulation

A. Network Model

To characterize the correlated and different sized networks, we assume that there is an undirect and acyclic underlying social network $G = \mathcal{G}(n, \mathbf{p})$ which represents the underlying social relationships between network nodes. Here n is the number of nodes and $\mathbf{p} = \{p_{uv}\}$ is the set of edge existence probabilities, where p_{uv} is the probability that there is an edge between two nodes u, v $(\in \{1, 2, \ldots, n\})$ in network G.

We further assume that the adversaries could observe a sanitized network G_A, where users' identities are all available, and an anonymized network G_B, of which users' identities are unavailable for privacy issue. In reality, for instance, G_A can be the online social relationships on the Internet, while G_B manifests the offline communication records crawled by the adversaries. For a static realization of $G = \mathcal{G}(n, \mathbf{p})$, we establish the sanitized network G_A by sampling only the edge set of G with probability s, while the anonymized network G_B is generated by sampling the edge set of G with probability s

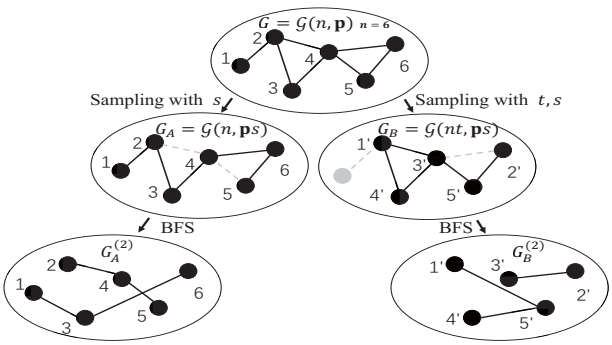


Fig. 1. The $\mathcal{G}(n, \mathbf{p}; t, s)$ network model composed of G, G_A ,G_B. Moreover, $G_A^{(2)}$ and $G_B^{(2)}$ are the derivative 2-hop adjacent networks.

and the node set of G with probability t. We note that the edges connecting removed nodes are also removed in the sampling process. As a result, the observed networks can be illustrated as $G_A = \mathcal{G}(n, \mathbf{p}s)$ and $G_B = \mathcal{G}(nt, \mathbf{p}s)$, with different but correlated node and edge sets. Here $\mathbf{p}s$ is a new set originated from multiplying s to each element in set $\mathbf{p}$. We can refer to Fig. 1 for the sampling process, where the gray node and the gray dashed edges are removed owing to sampling.

To sum up, we call the above model the $\mathcal{G}(n, \mathbf{p}; s, t)$ network model. Generally, a network model $\mathcal{G}(n, \mathbf{p}; s_1, s_2, t_1, t_2)$ can be composed of an underlying network $\mathcal{G}(n, \mathbf{p})$ and two observed networks formed by different edge sampling rates s_1, s_2 and node sampling rates t_1, t_2. However, when nodes u and v are only sampled in networks G_A and G_B, respectively, whether we match u with v or keep both of them unmatched will make no difference to the matching accuracy. Therefore, we focus on the simpler $\mathcal{G}(n, \mathbf{p}; s, t)$ model in the rest of the paper. In fact, as it will be clear in Section IV, even the analysis on this simplified model is non-trivial.

The one-hop relations between different user pairs in networks G, G_A and G_B can be represented by the adjacency matrices. However, as noted earlier, the adjacency matrices convey limited information for seedless de-anonymization, which motivates us to turn to multi-hop relations. To facilitate our discussion, we define some terminologies as follows.

Definition 1. ***(Multi-hop adjacency matrix)*** *Denote the node set of a network G_A as V_A. For nodes $u, v \in V_A$, the l-hop adjacency matrix $A^{(l)}$ follows that $A^{(l)}_{uv} = 1$ if and only if the shortest path between u and v is l and $A^{(l)}_{uv} = 0$ otherwise.*

Definition 2. ***(Multi-hop edge)*** *For nodes $u, v \in V_A$, we say there is an l-hop edge between them if $A^{(l)}_{uv} = 1$.*

Definition 3. ***(Multi-hop adjacent network)*** *The network with adjacency matrix $A^{(l)}$ is named as $G_A^{(l)}$. A network $G_A^{(l)}$ is an l-hop adjacent network if it is composed of l-hop edges.*

Obviously, $A^{(1)} = G_A$ and $G_A^{(1)} = G_A$. Hereinafter, the superscript (1) can be omitted without loss of readability. Besides, these definitions are also applicable for network G_B. Note that the multi-hop edges for each node can be detected through Breadth-First Searching (BFS). Fig. 1 exhibits a simple example of the $G(n, \mathbf{p}; s, t)$ model along with the 2-hop adjacent networks derived from G_A and G_B.

B. Problem Formulation of De-anonymization

1) The traditional de-anonymization with one-hop adjacency matrices. Given the adjacency matrices of social networks G_A and G_B, we have $A \in \mathbb{R}^{n_1 \times n_1}$ and $B \in \mathbb{R}^{n_2 \times n_2}$, where $\mathbb{R}$ is the set of real numbers, and n_1 and n_2 are the number of nodes in G_A and G_B, respectively. Without loss of generality, we can assume that $n_1 \geq n_2$. Mathematically, the social network de-anonymization problem can thus be formulated as

$$\boldsymbol{\mathcal{P}1}: \underset{P \in \Pi^{n_1 \times n_2}}{\text{minimize}} \|A - PBP^T\|_F^2, \tag{1}$$

where $P \in \Pi^{n_1 \times n_2}$ is an identify matrix satisfying $P \in \{0,1\}^{n_1 \times n_2}$, $P\mathbf{1}_{n_2} \preceq \mathbf{1}_{n_1}$ and $P^T\mathbf{1}_{n_1} = \mathbf{1}_{n_2}$. Here $\mathbf{1}_n$ is an $n \times 1$ vector of all ones, and $X \preceq X'$ means that their difference $X' - X$ is positive semi-definite. The matrix P can also be described as an injective function $\pi : V_A \to V_B$, where $\pi(u) = v$ if $P_{uv} = 1$ for any v and $\pi(u) = 0$ otherwise, for $u \in \{1, 2, \ldots, n_1\}$ and $v \in \{1, 2, \ldots, n_2\}$. Both forms ($P$ and π) will be utilized in the following sections.

2) The collective-form de-anonymization with multi-hop adjacency matrices. The de-anonymization accuracy is defined as the fraction of correctly matched nodes in the anonymized network G_B. However, as existing works [17], [18] have stated, the correct matching achieves the minimum of problem $\boldsymbol{\mathcal{P}1}$ only when the mean degree is $\Omega(\log n)$. When the networks get sparser, the solution for $\boldsymbol{\mathcal{P}1}$ may not work well. To make use of richer structural information, we collect the l-hop adjacent matrices $A^{(l)}$ and $B^{(l)}$ of G_A and G_B, as defined in Definition 1. Based on that, we can denote

$$\Delta_\pi^{(l)} = f^{(l)}(P) \coloneqq \|A^{(l)} - PB^{(l)}P^T\|_F^2 \tag{2}$$

as the adjacency disagreement between $A^{(l)}$ and $B^{(l)}$ under the injective function π (or the identify matrix P), and define the collective adjacency disagreement as follows.

Definition 4. ***(Collective adjacency disagreement)*** *The collective adjacency disagreement (CAD) at level l between networks G_A and G_B is the summation of the disagreements of their multi-hop adjacency matrices within l hops, i.e.,*

$$\Gamma_\pi^{(l)} = g^{(l)}(P) \coloneqq \sum\nolimits_{i=1}^{l} f^{(i)}(P). \tag{3}$$

In the following, we transform our overarching goal to solve the collective-form de-anonymization problem as

$$\boldsymbol{\mathcal{P}2}: \underset{P \in \Pi^{n_1 \times n_2}}{\text{minimize}} \; g^{(l)}(P). \tag{4}$$

Remark. Since we regard the minimal disagreements as our goal, the existence of structurally equivalent nodes will hinder the de-anonymization process, as they are indistinguishable under our target function. However, the whole network can be correctly matched under certain conditions, which, in other words, ensure that every node can be unique and identifiable.

IV. Conditions for Correct Matching

With the network model and the formulated problem, we derive the theoretical conditions for correctly matching two observed networks in this section. To proceed in an orderly way, we first explore the condition for correct matching in a special case of $l = 1$, then extend the result to $l \geq 1$.

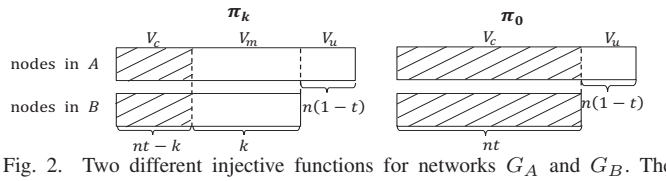


Fig. 2. Two different injective functions for networks G_A and G_B. The shaded parts are the set of correctly matched nodes V_c, while the blank parts are composed of the sets of mismatched nodes V_m and unmatched nodes V_u.

A. Preliminary Divisions of the Nodes and the Node Pairs

For a given injective function π, the nodes in different networks can be classified into three categories: correctly matched ($v \in V_c$), mismatched ($v \in V_m$), and unmatched ($v \in V_u$), as illustrated in Fig. 2. Considering that the number of unmatched vertices is $|V_u| = n(1-t)$ when n is large, we can further denote the injective function as π_k if there are exactly k mismatched vertices (on the left side of Fig. 2). Therefore, π_0 (on the right side of Fig. 2) stands for the correct matching where all nodes in B are correctly matched.

Accordingly, we can divide the node pairs [1] into three parts. The first part, denoted as $E_{c,k}$, is the correctly matched pairs where both nodes are from V_c; the second part is denoted as $E_{m,k}$, which is the mismatched node pairs where one of them is a mismatched node and the other is either mismatched or correctly matched; the last part is the unmatched pairs denoted as E_u, at least one node of whom is unmatched. The expected cardinalities of $E_{c,k}$, $E_{m,k}$ and E_u can be expressed as

$$\begin{cases} |E_{c,k}| = \binom{nt-k}{2}, \\ |E_{m,k}| = \binom{k}{2} + \binom{k}{1}\binom{nt-k}{1}, \\ |E_u| = \binom{n(1-t)}{2} + \binom{n(1-t)}{1}\binom{nt}{1}. \end{cases} \tag{5}$$

B. Condition for Correct Matching With $l = 1$

Based on the definitions above, we provide the condition under which correct matching is achievable by minimizing the adjacency disagreement Δ_π, as stated in Theorem 1. To clarify, we denote $p_{\max} = \max(\mathbf{p})$ as the maximum in set $\mathbf{p}$ and $\bar{p} = \mathbb{E}[\mathbf{p}]$ as the expectation of all elements in set $\mathbf{p}$.

Theorem 1. *For the network model $\mathcal{G}(n, \mathbf{p}; s, t)$ with observed networks G_A, G_B, the correct injective function π_0 minimizes the adjacency disagreement $\Gamma_\pi^{(1)} = \Delta_\pi$ with probability going to 1 as the number of nodes n goes to infinity, if sampling rates $s, t = \Theta(1)$, while $p_{\max} = o(1)$ and*

$$\bar{p} = \frac{12(2-s)}{s^3 t}\frac{(\log n)}{n} + \omega\left(\frac{1}{n}\right) = \Omega\left(\frac{\log n}{n}\right). \tag{6}$$

Proof. Let us take Fig. 2 again for illustration. We denote the adjacency disagreements of the two different injective functions π_k and π_0 as Δ_k and Δ_0, respectively. The outline of this proof can be divided into two steps: 1) For a given injective function π_k where exactly k nodes are wrongly matched, we approximate the probability of $\Delta_k \le \Delta_0$; 2) we then evaluate the number of injective functions π_k which may have lower adjacency disagreement than the correct matching, and prove that the expected errors for all these wrong matches are negligible under the conditions stated in the theorem.

1) The probability of $\Delta_k \le \Delta_0$. As node pairs in $E_{c,k}$ and E_u contribute equally to Δ_k and Δ_0, we write

$$\Delta_k - \Delta_0 = X_k - Y_k, \tag{7}$$

[1]Node pairs uv and vu will be the same pair regardless of their order.

where X_k is the number of wrongly matched edges in $E_{m,k}$ by π_k, Y_k is the number of edges in $E_{m,k}$ that are only sampled in G_A or G_B. They can be described as

$$\begin{cases} X_k = \sum_{uv\in E_{m,k}} |\mathbf{1}_{\{A_{uv}=1\}} - \mathbf{1}_{\{B_{\pi(u)\pi(v)}=1\}}|, \\ Y_k = \sum_{uv\in E_{m,k}} |\mathbf{1}_{\{A_{uv}=1\}} - \mathbf{1}_{\{B_{uv}=1\}}|. \end{cases} \tag{8}$$

Note that if $E_{m,k}$ is non-empty (only when $k > 0$), X_k can hardly equal to Y_k. We can then compare their expectations.

Since $P(A_{uv} = 1, B_{uv} = 0) = p_{uv}s(1-s) = p_{uv}s(1-s)$ and $P(A_{uv} = 0, B_{uv} = 1) = p_{uv}s(1-s)$, we can evaluate the expectation of Y_k as

$$\mathbb{E}[Y_k] = \mathbb{E}\left[\sum_{uv\in E_{m,k}} 2p_{uv}s(1-s)\right] = |E_{m,k}| \cdot 2\bar{p}s(1-s), \tag{9}$$

where $\bar{p} = \mathbb{E}[\mathbf{p}]$ is the expected value of edge existence probability in underlying graph G.

For X_k, if $uv \neq \pi(u)\pi(v)$, the probability that only one of the two edges is sampled is $p_{uv}s(1 - p_{\pi(u)\pi(v)}s) + p_{\pi(u)\pi(v)}s(1 - p_{uv}s)$. Meanwhile, there could be at most $k/2$ node pairs that make $uv = \pi(u)\pi(v)$ with the fact that $|V_m| = k$. Since $k/2 \ll |E_{m,k}|$, we approximate the expectation of X_k as

$$\begin{aligned} \mathbb{E}[X_k] =& \mathbb{E}\left[\sum_{uv\in E_{m,k}} p_{uv}s + p_{\pi(u)\pi(v)}s - 2p_{uv}p_{\pi(u)\pi(v)}s\right] \\ =& |E_{m,k}| \cdot 2\bar{p}s. \end{aligned} \tag{10}$$

Here $p_{uv}p_{\pi(u)\pi(v)} (\ll p_{uv})$ are omitted in the approximation.

According to Lemma A.1 in [20], for two random variables X and Y which are the sum of independent Bernoulli variables with $\mathbb{E}[X] \ge \mathbb{E}[Y]$, the probability of $X - Y \le 0$ satisfies

$$P(X - Y \le 0) \le 2\exp\left(\frac{-(\mathbb{E}[X] - \mathbb{E}[Y])^2}{12(\mathbb{E}[X] + \mathbb{E}[Y])}\right). \tag{11}$$

Hence, we approximate the probability of $\Delta_k - \Delta_0 \le 0$ by

$$\begin{aligned} P(\Delta_k - \Delta_0 \le 0) &= P(X_k - Y_k \le 0) \\ &\le 2\exp\left(\frac{-(\mathbb{E}[X_k] - \mathbb{E}[Y_k])^2}{12(\mathbb{E}[X_k] + \mathbb{E}[Y_k])}\right) \\ &\approx 2\exp\left(-|E_{m,k}| \cdot \frac{\bar{p}}{6}\frac{s^3}{2-s}\right) \\ &\approx 2\exp\left(-k(2nt-k) \cdot \frac{\bar{p}}{12}\frac{s^3}{2-s}\right). \end{aligned} \tag{12}$$

2) The expected number of wrong matches S. We denote $S_k = \sum_{\pi_k} \mathbf{1}_{\{\Delta_k \le \Delta_0\}}$ as the number of wrong matches with k mismatched nodes that has lower adjacency disagreement than the correct matching. Therefore, the expected number of total wrong matches $S = \sum_{k=1}^{nt} S_k$ that have smaller adjacency disagreement than π_0 is

$$\begin{aligned} \mathbb{E}[S] &= \sum_{k=1}^{nt} \mathbb{E}[S_k] = \sum_{k=1}^{nt}\sum_{\pi_k} \mathbb{E}[\mathbf{1}_{\{\Delta_k \le \Delta_0\}}] \\ &= \sum_{k=1}^{nt}\sum_{\pi_k} P(\Delta_k \le \Delta_0) \le \sum_{k=1}^{nt} n^k P(\Delta_k \le \Delta_0) \\ &\le 2\sum_{k=1}^{nt} n^k \exp\left(-k(2nt-k) \cdot \frac{\bar{p}}{12}\frac{s^3}{2-s}\right) \\ &\le 2\sum_{k=1}^{nt} \exp\left(k\left(\log n - \frac{n\bar{p}}{12}\frac{s^3 t}{2-s}\right)\right). \end{aligned} \tag{13}$$

Since we have assumed that $\bar{p} = \frac{12(2-s)}{s^3 t}\frac{\log n}{n} + \omega\left(\frac{1}{n}\right)$ in the

statement of Theorem 1, the first term of this summation goes to zeros, whereby the whole summation goes to zero. □

Remark. In the derivation, we approximate the summation of edge existence probabilities for node pairs in E_m, with the global average existence probability $\bar{p}$, which proves the Theorem 1 only in a statistical sense. Put differently, there could be a small set of special cases that satisfy our conditions but cannot be correctly matched. This problem can be fixed if the expectation $\bar{p}$ is replaced with $p_{\min} = \min(\mathbf{p})$ in the statement of this theorem. Besides, Eqn. (13) suggests that only a limited number S of wrong matches may have smaller adjacency disagreements than the correct matching. Hence, the probability that we find the correct matching can be approximated by $1 - S/n!$, where $n!$ is the number of all possible permutations with n nodes.

C. General Condition for Correct Matching With $l \geq 1$

Supported by Theorem 1, we can further expose similar conditions for $l \geq 1$, and state our result in Theorem 2.

Theorem 2. *For the network model $\mathcal{G}(n, \mathbf{p}; s, t)$ with observed networks G_A, G_B, denote $p_{\max} = \max(\mathbf{p})$ and $\bar{p} = \mathbb{E}[\mathbf{p}]$. The correct injective function π_0 minimizes the collective adjacency disagreement $\Gamma_\pi^{(l)}$ with probability going to 1 as the number of nodes n goes to infinity, if the hops $l = \Theta(1)$, sampling rates $s, t = \Theta(1)$, $p_{\max} = o\left(n^{1/l}/n\right)$, and*

$$\bar{p} = \frac{12(1 + t^{l-1} - s^l t^{l-1})}{s^{3l} t^{2l-1}} \frac{(\log n)^{1/l}}{n} + \omega\left(\frac{1}{n}\right) = \Omega\left(\frac{(\log n)^{1/l}}{n}\right).$$

Proof. Let us take a look back at Fig. 1, where we say there is an l-hop edge between nodes u and v in network G_A if $A_{uv}^{(l)} = 1$. Note that the three different node sets (V_c, V_m, V_u) will stay invariant for l-hop adjacency matrices $A^{(l)}$ and $B^{(l)}$, with the corresponding node pairs remaining unchanged. Recall that the definition of collective adjacency disagreement $\Gamma_\pi^{(l)}$ is

$$\Gamma_\pi^{(l)} = \sum\nolimits_{i=1}^{l} \Delta_\pi^{(i)}. \tag{14}$$

Essentially, we mean to minimize the sum of $\Delta_\pi^{(l)}$ with different number of hops l. Similar to Eqn. (8), we can define $X_k^{(l)}$ as the number of wrongly matched multi-hop edges within l hops in $E_{m,k}$ under injective function π_k, and $Y_k^{(l)}$ as the number of multi-hop edges within l hops in $E_{m,k}$ under the correct injective function π_0. They are expressed as

$$\begin{cases} X_k^{(l)} = \sum_{uv \in E_{m,k}} \sum_{i=1}^{l} |\mathbf{1}_{\{A_{uv}^{(i)}=1\}} - \mathbf{1}_{\{B_{\pi(u)\pi(v)}^{(i)}=1\}}|, \\ Y_k^{(l)} = \sum_{uv \in E_{m,k}} \sum_{i=1}^{l} |\mathbf{1}_{\{A_{uv}^{(i)}=1\}} - \mathbf{1}_{\{B_{uv}^{(i)}=1\}}|. \end{cases} \tag{15}$$

For brevity, we can simply write Γ_{π_k} by Γ_k. By the fact that

$$\Gamma_k^{(l)} - \Gamma_0^{(l)} = X_k^{(l)} - Y_k^{(l)}, \tag{16}$$

we can approximate the probability of $\Gamma_k^{(l)} < \Gamma_0^{(l)}$ with Eqn. (11) and the results of Lemma 1, whose detailed derivation is deferred to Appendix A. We further denote $S_k^{(l)} = \sum_{\pi_k} \mathbf{1}_{\{\Gamma_k^{(l)} \leq \Gamma_0^{(l)}\}}$ as the number of wrong matches that have exactly k mismatched nodes and lower CAD at level l than that of the correct matching, and $S^{(l)} = \sum_{k=1}^{nt} S_k^{(l)}$ as the summation of them. In the vein of inequalities (12) and (13), we calculate the upper bound of the expectation of $S^{(l)}$ by

$$\mathbb{E}[S^{(l)}] \leq 2 \sum_{k=1}^{n} \exp\left(k\left(\log n - \frac{n^l \bar{p}^l}{12} \frac{s^{3l} t^{2l-1}}{1 + t^{l-1} - s^l t^{l-1}}\right)\right).$$

As long as $\bar{p} = \Omega\left(\frac{(\log n)^{1/l}}{n}\right)$ and $n \to \infty$, the term $\log n - \frac{n^l \bar{p}^l}{12} \frac{s^{3l} t^{3l-2}}{1 + t^{l-1} - s^l t^{l-1}} = -\omega(1)$. Further, once the first term of this summation goes to 0, the whole summation also goes to 0, which leads to $E[S^{(l)}] \to 0$. This completes our proof. □

Lemma 1. *When $n \to \infty$, $p_{max} = o(n^{1/l}/n)$ and $\bar{p} = \Omega\left((\log n)^{1/l}/n\right)$, the expectations of $X_k^{(l)}$ and $Y_k^{(l)}$ can be approximated as*

$$\begin{cases} \mathbb{E}[X_k^{(l)}] = |E_{m,k}|\left(n^{l-1} \bar{p}^l s^l (1 + t^{l-1})\right), \\ \mathbb{E}[Y_k^{(l)}] = |E_{m,k}|\left(n^{l-1} \bar{p}^l s^l (1 + t^{l-1} - 2 s^l t^{l-1})\right). \end{cases} \tag{17}$$

Remark. To establish an intuitive understanding of Theorem 2, we may approximate the maximal degree for nodes in a network as $k_{\max} = n p_{\max}$ and the mean degree as $\bar{k} = n\bar{p}$. The conditions for edge existence probabilities $p_{\max}$ and $\bar{p}$ in our theorem can be transformed to the conditions that $k_{\max} = o(n^{1/l})$ and $\bar{k} = \Omega((\log n)^{1/l})$. As the mean degree for this network is $\Omega((\log n)^{1/l})$, a node will averagely get $\Omega(\log n)$ l-hop neighbors, which reaches the information-theoretic lower bound [25] to correctly match two networks.

On this basis, Eqn. (17) implies the fact that the expected errors caused from wrongly matched pairs are larger than those from correctly matched pairs ($\mathbb{E}[X_k^{(l)}] > \mathbb{E}[Y_k^{(l)}]$), and the difference between the two errors gets larger as l increases ($\mathbb{E}[X_k^{(l)}] - \mathbb{E}[Y_k^{(l)}] = \Theta(n^{l-1} p^l)$ is positively correlated with l). This also interprets that higher level of collective adjacency disagreement can be more beneficial to the de-anonymization of sparser networks as the number of nodes n is large enough.

V. Algorithms

While Theorem 2 ensures the possibility of correct matching, its algorithmic realization still remains to be unsolved. In this section, we design a collective de-anonymization algorithm (CDA), and refine it in both efficiency and effectiveness to match two networks that contain node sets of different sizes.

A. Collective De-anonymization Algorithm

Recall that $\boldsymbol{\mathcal{P}2}$ (Eqn. (4)) is intrinsically a combinatorial optimization problem with a discrete feasible region. The combinatorial nature of the feasible region determines that finding a global optimum of $\boldsymbol{\mathcal{P}2}$ is NP-hard [8]. Thus, instead of directly solving this problem, we first rewrite Eqn. (2) as

$$\begin{aligned} \Delta_\pi^{(l)} &= \|A^{(l)} - P B^{(l)} P^T\|_F^2 \\ &= \|A^{(l)}\|_F^2 + \|B^{(l)}\|_F^2 - 2\mathrm{tr}\left(A^{(l)} P B^{(l)} P^T\right). \end{aligned} \tag{18}$$

Further, by denoting

$$h^{(l)}(P) = \sum\nolimits_{i=1}^{l} \mathrm{tr}\left(A^{(i)} P B^{(i)} P^T\right), \tag{19}$$

the problem $\boldsymbol{\mathcal{P}2}$ can be equivalent to

$$\boldsymbol{\mathcal{P}3}: \underset{P \in \Pi^{n_1 \times n_2}}{\text{maximize}}\ h^{(l)}(P). \tag{20}$$

Classically, we can then relax the feasible region of P from the discrete $\Pi^{n_1 \times n_2}$ to a continuous $\mathcal{D}^{n_1 \times n_2}$. For $P \in \mathcal{D}^{n_1 \times n_2}$, it follows $P \in [0,1]^{n_1 \times n_2}, P\mathbf{1}_{n_2} \preceq \mathbf{1}_{n_1}, P^T \mathbf{1}_{n_1} = \mathbf{1}_{n_2}$. Such relaxation ensures the feasible region is continuous and convex, thereby bringing the following convex problem:

$$\boldsymbol{\mathcal{P}4}: \underset{P \in \mathcal{D}^{n_1 \times n_2}}{\text{maximize}}\ h^{(l)}(P). \tag{21}$$

Algorithm 1 Collective De-anonymization Algorithm

Input: Adjacent matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$ $(m \le n)$; number of hop l; (optional) matrix $P \in \mathcal{D}^{n\times m}$.
Output: The optimal injective function $\pi : V \to V$.

1: **Initialize:** $P = \frac{\mathbf{1}_n \cdot \mathbf{1}_m^T}{n}$ if it is not specified; $A^{(i)}$ and $B^{(i)}$ $(i \le l)$ computed from A and B with Breadth-First Searching; step size $\alpha = 1$; step tolerance $\epsilon = 10^{-4}$.
2: **while** $\alpha > \epsilon$ **do**
3: Compute $\nabla h^{(l)}(P)$ from Eqn. (22);
4: $Q = \arg\max_{Q\in\Pi_{n\times m}} \text{tr}(Q^T \nabla h^{(l)}(P))$;
5: Compute step size $\alpha = \arg\max_\alpha h^{(l)}((1-\alpha)P + \alpha Q)$ over $\alpha \in [0, 1]$;
6: Renew the matrix $P = (1-\alpha)P + \alpha Q$;
7: Compute $P^* = \arg\max_{Q\in\Pi^{n\times m}} \text{tr}(Q^T P)$;
8: **return** π, where $\pi(u) = v$ if $P^*_{uv} = 1$ for any v and $\pi(u) = 0$ if $P^*_{uv} = 0$ for all v.

Thereafter, the solution for the convex problem $\mathcal{P}\mathbf{4}$ can be approached with the Frank-Wolfe method [26], [8], which iteratively minimizes the linear approximation of the objective function given by its first-order Taylor approximation and moves towards a minimizer of this linear function. We thus provide the gradient of $h^{(l)}(P)$ as follows.

$$\nabla h^{(l)}(P) = 2\sum\nolimits_{i=1}^{l} A^{(i)} P B^{(i)}. \tag{22}$$

With the gradient of $h^{(l)}(P)$, we can exhibit our CDA as Algorithm 1. The main idea of this algorithm inherits Frank-Wolfe method (lines 2-6). Besides, the convex problems at lines 4 and 7 are solved by Hungarian Algorithm [27], which is one of the most popular combinatorial optimization algorithm that solves the assignment problem within $O(n^3)$ time complexity. The complexity of the whole algorithm is also $O(n^3)$ because the iteration of the Frank-Wolfe method terminates in finite steps. Since the Frank-Wolfe method does not require the objective function to be square matrix, our algorithm is also suitable for two graphs of differently sized vertex sets. The multi-hop adjacency matrices $A^{(i)}$ and $B^{(i)}$ $(i \le l)$ are also utilized in the computation of $h^{(l)}(P)$.

B. The Refined Algorithm

We further refine Algorithm 1 with respect to both efficiency and effectiveness. On the one hand, as the Hungarian algorithm costs too much time, we incorporate the Sinkhorn method [28], which alternately re-scales all rows and all columns of a matrix to sum to 1 and costs $O(n^2)$, to approximate the solution of the convex problem in the iteration. On the other hand, we presort the nodes in both networks by degree before matching, which can narrow the distance between to-be-matched nodes from the intuition that nodes are likely to be matched with those who have similar degree [9]. The refined collective de-anonymization algorithm (RCDA) can be referred as Alg. 2.

In Alg. 2, the Sinkhorn function at line 4 can normalize the summation of the rows and columns of the input matrix, which works as illustrated in Alg. 3. By taking exponential on $h^{(l)}(P)$, the input matrix $\exp(h^{(l)}(P))$ is positive definite that satisfies the requirement of Sinkhorn method. The output, i.e., the normalized matrix, is an approximation to the solution of the original problem at line 4 in Alg. 1. The first six lines in

Algorithm 2 Refined Collective De-anonymization Algorithm

Input: Adjacent matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$ $(m \le n)$; number of hop l; (optional) matrix $P \in \mathcal{D}^{n\times m}$.
Output: The optimal injective function $\pi : V \to V$.

1: **Initialize:** $P = \frac{\mathbf{1}_n \cdot \mathbf{1}_m^T}{n}$ if not specified; A and B sorted in a decreasing order by the sum of their column; $A^{(i)}$ and $B^{(i)}$ $(i \le l)$ computed from A and B with Breadth-First Searching; step size $\alpha = 1$; step tolerance $\epsilon = 10^{-4}$.
2: **while** $\alpha > \epsilon$ **do**
3: Compute $\nabla h^{(l)}(P)$ from Eqn. (22);
4: $Q = \text{Sinkhorn}(\exp\left(\nabla h^{(l)}(P)\right))$;
5: Compute step size $\alpha = \arg\max_\alpha h^{(l)}((1-\alpha)P + \alpha Q)$ over $\alpha \in [0, 1]$;
6: Renew the matrix $P = (1-\alpha)P + \alpha Q$;
7: Compute $P^* = \arg\max_{Q\in\Pi^{n\times m}} \text{tr}(Q^T P)$;
8: **return** $\pi = \text{CDA}(A, B, l, P^*)$.

Algorithm 3 Sinkhorn

Input: A positive definite matrix $P \in \mathbb{R}^{n\times m}$ with $n > m$.
Output: A matrix $Q \in \mathcal{D}^{n\times m}$.

1: **Initialize:** $Q = \mathbf{0}^{n\times m}$; tolerance $\epsilon = 10^{-16}$.
2: **while** $\|P - Q\|_F^2 > \epsilon$ **do**
3: normalize across rows by $Q_{uv} = P_{uv} / \sum_{u=1}^{n} P_{uv}$;
4: normalize across columns by $P_{uv} = Q_{uv} / \sum_{v=1}^{m} Q_{uv}$;
5: **return** Q.

Alg. 2 cost $O(n^2)$ since the iteration (lines $2-6$) usually ends in finite rounds. It is also worth noting that the last line of Alg. 2 calls Alg. 1 once with the computed permutation matrix P^* as a supplementary input parameter. Since the approximated method decreases the matching accuracy, a call for Alg. 1 compensates the accuracy loss with slightly more time costs. Nevertheless, as we have obtained a roughly accurate P^* at line 7 as preliminary, the complexity of Alg. 2, though remains $O(n^3)$ in order sense, will lead to much shorter running time in practice than that of directly calling for Alg. 1.

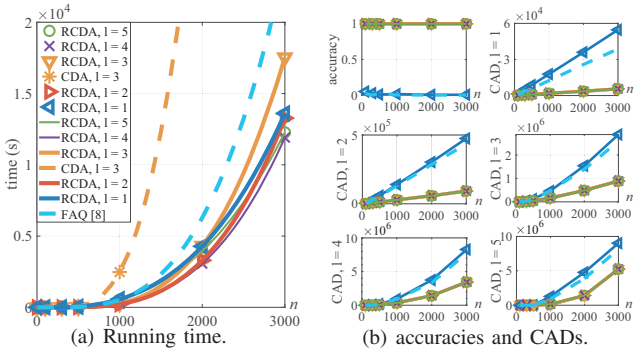
Remark. A family of literature has paid endeavor to the exploration of heuristic algorithms for social network de-anonymization. As far as we concern, our work is an initial attempt to incorporate the collectiveness into existing heuristics. Though the proposed algorithms in this section are based on a simple heuristic, the power of collectiveness, as it will be clear, brings significant improvements. Our qualitative finding is expected to hold in future combination with other heuristics.

VI. Experiments

We proceed to evaluate the performance of our proposed RCDA algorithm in this section. Since we are matching two networks in the proposed algorithm, we use the terms 'de-anonymization process' and 'the matching process' alternatively. We evaluate the running time, matching accuracy and the collective adjacency disagreement (CAD).

A. Synthetic Networks

Synthetic experiments are performed on both Erdős-Rényi (ER) networks and Scale-Free (SF) networks. For an ER network $\mathcal{G}(n, p)$, every node pair can be connected with the same probability p. For an SF network $\mathcal{G}(n, \lambda, c)$, its degree distribution follows $d(k) \sim k^{-\lambda}$ and $\mathbb{E}[d(k)] = c$.



(a) Running time. (b) accuracies and CADs.

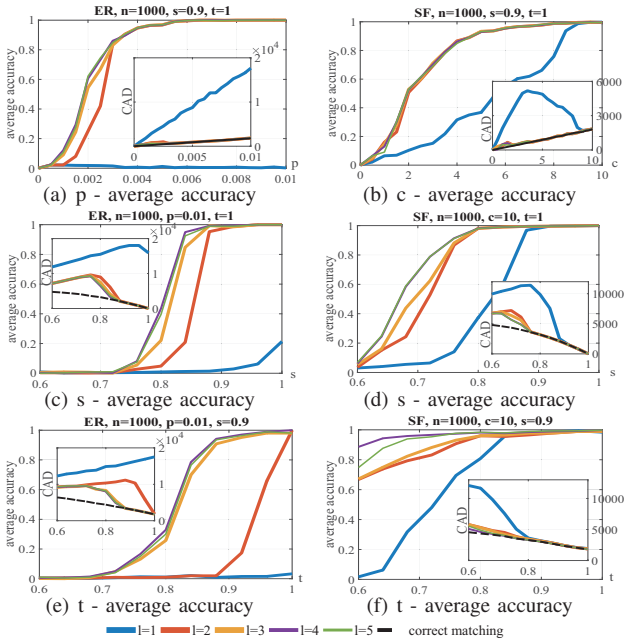
Fig. 3. (a) comparison of the running time for RCDA with different collective number of hops l and the comparison between RCDA and CDA; (b) the accuracy and collective adjacency disagreements (CADs) at levels 1 to 5.

In the sequel, we first evaluate the running time of the proposed algorithm and compare the collective adjacency disagreements at different levels. To this end, we build different ER networks where the cardinality of the node sets varies from 100 to 3,000. Other parameters are set to be $p = \log n/n$, $s = 0.9$ and $t = 1$. In Fig. 3, the points are the averagely costed time in 10 experiments under given circumstances, and the curves are the fitted curves for them. In Fig. 3(a), the time cost function for RCDA with $l = 3$ (the full curve in yellow) can be approximated by $t(n) = 9.1 * 10^{-7}n^3 - 8.5 * 10^{-4}n^2 + 0.23n - 7.1$, while the time cost function for CDA with $l = 3$ (the dashed curve in yellow) can be approximated by $t(n) = 6.8 * 10^{-6}n^3 - 5.5 * 10^{-3}n^2 + 1.2n - 33$. This confirms our theoretical conclusion that the complexity for both algorithms are $O(n^3)$. Meanwhile, the refined algorithm does save a large amount of running time in that sense.

We also implement FAQ algorithm [8] in this figure since it is a special case of the CDA algorithm when $l = 1$. It can be observed that it yields similar performance to RCDA with $l = 1$. Both algorithms fail to de-anonymize the network when $l = 1$. Further, the RCDA algorithm costs similar time when it makes use of different collective levels l. This is because it only costs $O(nl)$ to find the l-hop neighbors for all nodes with the Breath-First Searching method. We should also note that that higher level of collectiveness does not necessarily mean higher efficiency or accuracy, as the networks in simulations do not have infinite number of nodes. When there is enough information to realize the correct matching, extra information may only cost extra time without performance improvement.

The results in Fig. 3(b) are two-folded: 1) when we do not utilize the collectiveness (FAQ and RCDA with $l = 1$), the matching accuracies approach 0 and the CADs get large; 2) when the collectiveness are considered (RCDA with $l \geq 2$), the accuracies become 1 and the CADs approach their minimum. The comparison among sub-figures in Fig. 3(b) also demonstrates a commonality that when networks can be correctly matched, CADs at different levels achieve their minima. Therefore, for simplicity, we plot only the CAD at level 1 (i.e., the adjacency disagreement) in the following simulations on both ER and SF networks if not specified.

We then compare the performance of the proposed RCDA algorithm with different parameters on both ER networks and



(a) p - average accuracy (b) c - average accuracy
(c) s - average accuracy (d) s - average accuracy
(e) t - average accuracy (f) t - average accuracy

Fig. 4. Simulations on ER and SF networks.

SF networks. The elementary setting for these models is: the number of nodes $n = 1000$, mean degree of the underlying network $c = 10$, edge sampling rate $s = 0.9$ and node sampling rate $t = 1$. We then tune the mean degree and sampling rates as illustrated in Fig. 4. We take an average of 100 times of duplicate experiments for each parameter setting to eliminate the side effects from sampling randomness and figure out the commonness in different structured networks.

1) On the mean degree: Figs. 4(a) and 4(b) exhibit the experiments on a sequence of mean degree c of the underlying network. For ER networks, the edge existence probability p corresponds to the ratio of the mean degree and the number of nodes. As the figures display, with the help of collectiveness, the matching accuracy gets higher with the increase of mean degree, and approaches 1 when the mean degree is larger than $\log n = 6.9$. In fact, a large fraction of nodes are correctly de-anonymized with collectiveness even when $c = 2$, where the synthetic networks are quite sparse. It is also worth noting that the adjacency disagreements are almost as small as those of the correct matching when $l \geq 2$, which suggests that these achieved accuracies are the best matching performance if the collective adjacency disagreement is given as the cost function.

2) On the edge sampling rate: Figs. 4(c) and 4(d) plot the results for varied edge sampling rates s on both observed networks. Since the mean degree of the observed networks arises from the multiplication of the mean degree of the underlying network and the edge sampling rate, the matching accuracy returns a similar trend to Figs. 4(a) and 4(b). The difference appears between their adjacency disagreements, which get increased with the growth of mean degree c but get decreased with the enlargement of edge sampling rate s. When the disagreements are compared with that of the correct matching, it is clear that this algorithm achieves minimal CADs only when s is large, where the accuracies approach 1.

This also indicates that the proposed algorithm can be further improved for better matching performance.

3) On the node sampling rate: Figs. 4(e) and 4(f) are illustrated by varying the node sampling rate t on the observed anonymized network. Since the random removal of nodes will not influence the mean degree to the observed networks in the average sense, the performance becomes different for ER networks and SF networks. Although some nodes may never be surely matched as they have same structure with others in the SF networks, just like the edge points in a star network, the RCDA algorithm brings high accuracies in a wide range. As we can find from Fig. 4(f), the accuracy is hardly decreased even when we remove 40% of nodes in the second Scale-Free network. This occurs because the degree of nodes in the SF network gets larger variance, which leads to a higher discernibility for these nodes.

Through a vertical comparison, we conclude that these three parameters in our test are equally influential to the performance of matching accuracy. In general, the larger these parameters are, the higher the matching accuracy will be.

B. Real Networks

We further perform experiments on the real-world Wikipedia Network [29] with 1382 nodes. The two observed networks are composed of an English version of 1382 entries collected in Wikipedia and a corresponding French version. An edge is created if there is a hyperlink in one entry that is directed to another. Since we have only the observed networks without a ground-truth, it is usually incapable for us to determine the edge sampling rate s. Instead, we can sample on the node set with rate t to form smaller observed networks. In consequence, we plot figures with varied node sampling rate t for the real datasets, where $(1-t)$ fraction of nodes in the French version network are randomly removed and then the remaining nodes are matched with those in the English version network.

The experimental result is reported in Fig. 5. Since this network cannot be fully correctly matched with our algorithm, we establish a thorough analysis on its collective adjacency disagreements at level 5.

This dataset performs the superiority of collectiveness. As it exhibits in Fig. 5, the experimental results can be divided into three categories: 1) For the RCDA algorithm with $l \geq 2$, a large fraction of nodes in this network can be correctly de-anonymized and their collective adjacency disagreements are relatively small; 2) for the RCDA algorithm with $l = 1$, it achieves about half of the accuracy of those who utilize collectiveness, while its CADs are even smaller than the CADs of others expect the correct matching in most cases; 3) for the FAQ algorithm (without collectiveness), almost no nodes can be correctly de-anonymized and its CADs are always higher than the CADs of the correct matching. This result is still a strong support to the power of collectiveness, which is claimed be helpful in improving the de-anonymization accuracy.

Then we focus on the transformation of accuracies and CADs along with the increase of the node sampling rate t. From this perspective, it can be also divided into three stages.

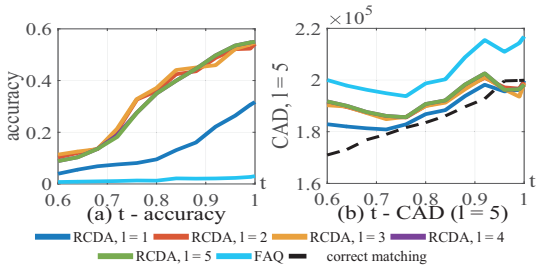


Fig. 5. Wikipedia data sets. One of the observed network is in English with $n = 1382$ entries, while the other is in French with nt $(0.6 \leq t \leq 1)$ entries.

For the first stage when $t < 0.7$, the CADs with different hop levels are far from that of the correct matching, thus they perform low accuracies. However, for the second stage, as the increase of t, especially when $0.7 < t < 0.95$, the CADs of the RCDA algorithms are close to the CAD of the correct matching. Consequently, the corresponding accuracies get steady rises. Finally when $t > 0.95$, these CADs become lower than the CAD of the correct matching, which, in result, brings few increases to the accuracies.

VII. Conclusion

In this paper, we introduce the collectiveness, i.e., a collection of multi-hop neighborhood relationships, into the social network de-anonymization problem. In theoretical aspect, we prove that sparse networks whose mean degrees are less than $\log n$ can also be correctly matched with the assist of collectiveness. On this basis, we propose the Refined Collective De-anonymization algorithm (RCDA), which maintains a time complexity of $O(n^3)$. From the experimental results, employing the collectiveness in this algorithm always provides higher matching accuracies and lower time costs than previous algorithms without collectiveness. Such performance improvements are applicable in differently structured networks, as well as in de-anonymizing the networks which possess different node set sizes with the sanitized network.

Acknowledgments

This work was supported by National Key R&D Program of China 2018YFB1004705, 2018YFB2100302 and by NSF China under Grant (No. 61822206, 61832013, 61960206002, 61532012).

Appendix A
Proof of Lemma 1

As defined, $X_k^{(l)}$ is the number of wrongly matched multi-hop edges within l hops in $E_{m,k}$ under the injective function $\pi = \pi_k$ while $Y_k^{(l)}$ is the number of multi-hop edges within l hops in $E_{m,k}$ under the correct injective function π_0, i.e.,

$$\begin{cases} X_k^{(l)} = \sum_{uv \in E_{m,k}} \sum_{i=1}^{l} \left| \mathbf{1}_{\{A_{uv}^{(i)}=1\}} - \mathbf{1}_{\{B_{\pi(u)\pi(v)}^{(i)}=1\}} \right|, \\ Y_k^{(l)} = \sum_{uv \in E_{m,k}} \sum_{i=1}^{l} \left| \mathbf{1}_{\{A_{uv}^{(i)}=1\}} - \mathbf{1}_{\{B_{uv}^{(i)}=1\}} \right|. \end{cases} \tag{23}$$

Note that the distance between each two different nodes u and v must be constant. We denote $d_G(u,v)$ as the shortest distance between nodes u and v in network G, i.e., $d_G(u,v) = l$ if and only if $G_{uv}^{(l)} = 1$, and denote

$$\begin{cases} x_{uv}^{(l)} := \sum_{i=1}^{l} \left| \mathbf{1}_{\{A_{uv}^{(i)}=1\}} - \mathbf{1}_{\{B_{\pi(u)\pi(v)}^{(i)}=1\}} \right|, \\ y_{uv}^{(l)} := \sum_{i=1}^{l} \left| \mathbf{1}_{\{A_{uv}^{(i)}=1\}} - \mathbf{1}_{\{B_{uv}^{(i)}=1\}} \right|, \end{cases} \tag{24}$$

which provides

$$X_k^{(l)} = \sum_{uv \in E_{m,k}} x_{uv}^{(l)}, \quad Y_k^{(l)} = \sum_{uv \in E_{m,k}} y_{uv}^{(l)}. \tag{25}$$

The parameters $x_{uv}^{(l)}$ and $y_{uv}^{(l)}$ can be re-written as

$$x_{uv}^{(l)} = \begin{cases} 2, & \text{if } d_A(u,v) \le l, d_B(\pi(u),\pi(v)) \le l, \\ & \quad \text{and } d_A(u,v) \ne d_B(\pi(u),\pi(v)), \\ 1, & \text{if } (d_A(u,v) \le l, d_B(\pi(u),\pi(v)) > l), \\ & \text{or } (d_A(u,v) > l, d_B(\pi(u),\pi(v)) \le l), \\ 0, & \text{if } (d_A(u,v) > l, d_B(\pi(u),\pi(v)) > l), \\ & \text{or } (d_A(u,v) = d_B(\pi(u),\pi(v)) \le l). \end{cases} \tag{26}$$

$$y_{uv}^{(l)} = \begin{cases} 2, & \text{if } d_A(u,v) \le l, d_B(u,v) \le l, \\ & \quad \text{and } d_A(u,v) \ne d_B(u,v), \\ 1, & \text{if } (d_A(u,v) \le l, d_B(u,v) > l), \\ & \text{or } (d_A(u,v) > l, d_B(u,v) \le l), \\ 0, & \text{if } (d_A(u,v) > l, d_B(u,v) > l), \\ & \text{or } (d_A(u,v) = d_B(u,v) \le l). \end{cases} \tag{27}$$

We then derive the expectations of $x_{uv}^{(l)}$ and $y_{uv}^{(l)}$ from the joint probability that nodes u, v are within distance l in both networks G_A and G_B.

For better illustration, we consider a special case where the underlying graph $G = \mathcal{G}(n,p)$ is an Erdős-Rényi random network [12], whose edges exist with same probability p. In network G, a node is expected to have $(n-1)p \approx np$ neighbors on average. Since we have assumed that $p = o(n^{1/l})/n$ and $l = \Theta(1)$, it could be intuitive (and is proved by [30]) that the expected number of l-hop neighbors for a node u in network G is $(np)^l (\ll n)$. In other words, the probability that the shortest distance between two certain nodes u and v is l can be approximated by

$$P(d_G(u,v) = l) = (np)^l / n = n^{l-1} p^l. \tag{28}$$

Further, since the distance between every two nodes is unique, it is reasonable to approximate that

$$P(d_G(u,v) \le l) = \sum\nolimits_{i=1}^{l} n^{i-1} p^i = n^{l-1} p^l. \tag{29}$$

The second equation establishes because we have assumed that $np \gg 1$, which indicates $n^{l-1}p^l \gg n^{i-1}p^i$ for all $i < l$.

Sampled from network G, networks $G_A = \mathcal{G}(n, ps)$ and $G_B = \mathcal{G}(nt, ps)$ are themselves random networks, thus

$$\begin{cases} P(d_A(u,v) = l) = n^{l-1}(ps)^l, \\ P(d_B(u,v) = l) = (nt)^{l-1}(ps)^l. \end{cases} \tag{30}$$

We denote $D_{uv}^{(l)}$ as the event that nodes u, v are of distance l in both G_A and G_B. If $d_G(u,v) = l$, the probability of $D_{uv}^{(l)}$ can be illustrated as the probability that there is at least one path from u to v in G that is sampled in network G_A and one path sampled in network G_B. Since the sampling process for networks G_A and G_B is independent from each other, we get

$$P(d_A(u,v) = l | d_G(u,v) = l) = s^l, \tag{31}$$

$$P(d_B(u,v) = l | d_G(u,v) = l) = s^l t^{l-1}, \tag{32}$$

$$P(D_{uv}^{(l)} | d_G(u,v) = l) = s^l \cdot s^l t^{l-1}. \tag{33}$$

On the contrary, if $d_G(u,v) \ne l$ especially when $d_G(u,v) < l$, it is complicated to find the probability of $D_{uv}^{(l)}$. Luckily, it can be asserted that $P(D_{uv}^{(l)} | d_G(u,v) < l) \le s^{l-1} \cdot s^{l-1} t^l$, as there should be at least one l-length path from u to v in G being sampled to networks G_A and G_B to make $D_{uv}^{(l)}$ true. Therefore,

$$\begin{aligned} P(D_{uv}^{(l)}) =& P(d_G(u,v) = l) P(D_{uv}^{(l)} | d_G(u,v) = l) \\ & + P(d_G(u,v) < l) P(D_{uv}^{(l)} | d_G(u,v) < l) \\ =& n^{l-1} p^l s^{2l} t^{l-1} \\ & + \left(\Sigma_{i=1}^{l-1} n^{i-1} p^i \right) P(D_{uv}^{(l)} | d_G(u,v) < l) \\ \overset{(*)}{=}& n^{l-1} p^l s^{2l} t^{l-1}. \end{aligned} \tag{34}$$

Here $(*)$ establishes because $n^{l-1}p^l$ is the dominating component in this equation.

Moreover, we can approximate the joint probability of $d_A(u,v) > l$ and $d_B(u,v) > l$ as follows.

$$\begin{aligned} & P(d_A(u,v) > l, d_B(u,v) > l) \\ =& P(d_G(u,v) > l) \cdot 1 + P(d_G(u,v) \le l) \\ & \cdot P(d_A(u,v) > l, d_B(u,v) > l | d_G(u,v) \le l) \\ =& (1 - n^{l-1}p^l) + n^{l-1}p^l \cdot (1 - s^l)(1 - s^l t^{l-1}) \\ =& 1 - n^{l-1} p^l s^l (1 + t^{l-1} - s^l t^{l-1}). \end{aligned} \tag{35}$$

With Eqns. (30), (34) and (35), we have

$$\begin{aligned} & P(y_{uv}^{(l)} = 0) \\ =& P(d_A(u,v) > l, d_B(u,v) > l) + \Sigma_{i=1}^{l} P(D_{uv}^{(i)}) \\ =& [1 - n^{l-1} p^l s^l (1 + t^{l-1} - s^l t^{l-1})] + \Sigma_{i=1}^{l} n^{l-1} p^l s^{2l} t^{l-1} \\ =& 1 - n^{l-1} p^l s^l (1 + t^{l-1} - 2 s^l t^{l-1}). \end{aligned} \tag{36}$$

Meanwhile,

$$\begin{aligned} & P(y_{uv}^{(l)} = 1) \\ =& P(d_A(u,v) > l, d_B(u,v) \le l) \\ & + P(d_A(u,v) \le l, d_B(u,v) > l) \\ =& P(d_B(u,v) \le l) P(d_A(u,v) > l | d_G(u,v) \le l) \\ & + P(d_A(u,v) \le l) P(d_B(u,v) > l | d_G(u,v) \le l) \\ =& (nt)^{l-1} (ps)^l (1 - s^l) + n^{l-1} (ps)^l (1 - s^l t^{l-1}) \\ =& n^{l-1} p^l s^l (1 + t^{l-1} - 2 s^l t^{l-1}). \end{aligned} \tag{37}$$

By calculation, $P(y_{uv}^{(l)} = 2) = 1 - P(y_{uv}^{(l)} = 0) - P(y_{uv}^{(l)} = 1) = 0$. We note that this probability should be a positive value in reality, while it is far less than $n^{l-1}p^l$ thus approximated to be zero here. Therefore, we can approximate the expectation

$$\mathbb{E}[y_{uv}^{(l)}] = \sum_{k=0}^{2} k \cdot P\left(y_{uv}^{(l)} = k\right) = n^{l-1} p^l s^l (1 + t^{l-1} - 2 s^l t^{l-1}). \tag{38}$$

In the vein of deriving $\mathbb{E}[y_{uv}^{(l)}]$, we can also obtain the expectation of $x_{uv}^{(l)}$ as

$$\mathbb{E}[x_{uv}^{(l)}] = n^{l-1} p^l s^l (1 + t^{l-1}). \tag{39}$$

According to Eqn. (25), the expectations of $X_k^{(l)}$ and $Y_k^{(l)}$ can thereupon be approximated with

$$\begin{aligned} \mathbb{E}[X_k^{(l)}] &= |E_{m,k}| \big(n^{l-1} p^l s^l (1 + t^{l-1}) \big), \\ \mathbb{E}[Y_k^{(l)}] &= |E_{m,k}| \big(n^{l-1} p^l s^l (1 + t^{l-1} - 2 s^l t^{l-1}) \big). \end{aligned} \tag{40}$$

Eqns. (28) to (40) are all derived based on an Erdős-Rényi network model. Finally, when it turns to an arbitrarily distributed network $\mathcal{G}(n, \mathbf{p})$, it is reasonable to replace the parameter p^l with the multiplication of a series of p_{uv}, i.e., $\prod_{i=1}^{l} p_{u_i v_i}$, since they represent the same probability (of l edges existing) in different models. This multiplication can be further approximated by $\bar{p}^l$, where $\bar{p} = \mathbb{E}[\mathbf{p}]$, considering that we assume p_{uv} for different node pairs u, v is independent from one another. Consequently, the expectations of $X_k^{(l)}$ and $Y_k^{(l)}$ can be approximated as we have stated in Eqn. (17).

REFERENCES

[1] D. Conte, P. Foggia, C. Sansone and M. Vento, "Thirty years of graph matching in pattern recognition", in *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 18, no. 3, pp. 265-298, 2004.

[2] L. Backstrom, C. Dwork and J. Kleinberg, "Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography", in *WWW*, 2007.

[3] A. Narayanan and V. Shmatikov, "De-anonymizing social networks", in *IEEE Symposium on Security and Privacy*, pp. 173-187, 2009.

[4] V. Lyzinski, D. E. Fishkind, M. Fiori, J. T. Vogelstein, C. E. Priebe and G. Sapiro, "Graph matching: relax at your own risk", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 60-73, 2016.

[5] L. Yartseva and M. Grossglauser, "On the performance of percolation graph matching", in *COSN*, pp. 119-130, Boston, Massachusetts, USA, Oct., 2013.

[6] C. F. Chiasserini, M. Garetto and E. Leonardi, "Social network de-anonymization under scale-free user relations", in *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3756-3769, 2016.

[7] T. Yu, J. Yan, Y. Wang, W. Liu and B. Li, "Generalizing graph matching beyond quadratic assignment model", in *NeurIPS*, Montréal, Canada, Dec., 2018.

[8] J. T. Vogelstein, et al., "Fast approximate quadratic programming for graph matching", in *PLOS One*, vol. 10, no. 4, art. no. e0121002, 2015.

[9] O. E. Dai, D. Cullina, N. Kiyavash and M. Grossglauser, "Analysis of a canonical labeling algorithm for the alignment of correlated Erdős-Rényi graphs", in *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, no. 2, art. no. 36, 2019.

[10] G. Zehfuss, "Ueber eine gewisse Determinante", in *Zeitschrift für Mathematik und Physik*, vol. 3, pp. 298-301, 1858.

[11] S. L. Feld, "Why your friends have more friends than you do", in *American Journal of Sociology*, vol. 96, no. 6, pp. 1464C1477, 1991.

[12] B. Bollobás, "Random graphs (2nd edition)", *Cambridge University Press*, Aug., 2001.

[13] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks", in *Reviews of Modern Physics*. vol. 74, no. 1, pp. 47-97, 2002.

[14] H. Amini and N. Fountoulakis, "Bootstrap percolation in power-law random graphs", in *Journal of Statistical Physics*, vol. 155, no. 1, pp. 72C92, 2014.

[15] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks", in *Proceedings of the VLDB Endowment*, vol. 7, no. 5, pp. 377C388, 2014.

[16] S. Nilizadeh, A. Kapadia and Y.-Y. Ahn, "Community-enhanced de-anonymization of online social networks",in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, Scottsdale, AZ, USA, Nov., 2014.

[17] P. Pedarsani and M. Grossglauser, "On the privacy of anonymized networks", in *Proc. ACM SIGKDD*, San Diego, California, USA, Aug., 2011.

[18] E. Kazemi, L. Yartseva and M. Grossglauser, "When can two unlabeled networks be aligned under partial overlap?", in *53rd Annual Allerton Conference*, Allerton House, UIUC, Illinois, USA, Sep., 2015.

[19] S. Ji, W. Li, M. Srivatsa and R. Beyah, "Structural data de-anonymization: quantification, practice, and implications", in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, Scottsdale, AZ, USA, Nov., 2014.

[20] L. Fu, X. Fu, Z. Hu, Z. Xu and X. Wang, "De-anonymization of social networks with communities: when quantifications meet algorithms", arXiv preprint arXiv:1703.09028v3, 2017.

[21] X. Wu, et al., "Social network de-anonymization with overlapping communities: analysis, algorithm and experiments", in *IEEE INFOCOM*, Honolulu, HI, USA, Apr., 2018.

[22] B. Nettasinghe, V. Krishnamurthy and K. Lerman, "Contagions in social networks: effects of monophilic contagion, friendship paradox and reactice networks", *arXiv preprint arXiv:1810.05822v1*, 2018.

[23] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation", in *Nature*, vol. 524, pp. 65-68, 2015.

[24] E. Mossel and J. Xu, "Seeded graph matching via large neighborhood statistics", *arXiv preprint arXiv:1807.10262v1*, 2018.

[25] D. Cullina and N. Kiyavash, "Improved achievability and converse bounds for Erdős-Rényi graph matching", in *ACM SIGMETRICS*, pp. 63-72, 2016.

[26] M. Frank, P. Wolfe, "An algorithm for quadratic programming", in *Naval Research Logistics Quarterly*, vol. 3, pp. 95-110, 1956.

[27] H. W. Kuhn, "The Hungarian method for the assignment problem", in *Naval Research Logistics*, vol. 2, pp. 83-97, 1955.

[28] M. Cuturi, "Sinkhorn distance: lightspeed computation of optimal transport", in *NeurIPS*, Lake Tahoe, Nevada, Dec., 2013.

[29] D. E. Fishkind, et al., "Seeded graph matching", in *Pattern Recognition*, vol. 87, pp. 203-215, 2019.

[30] R. Durrett, "Random graph dynamics", *Cambridge University Press*, 2007.