Differentially Private Community Detection in Attributed Social Networks

Tianxi Ji[*] txj116@case.edu
Changqing Luo[+] cluo@vcu.edu
Yifan Guo[*] yxg383@case.edu
Jinlong Ji[*] jxj405@case.edu
Weixian Liao[†] wliao@towson.edu
Pan Li[*] pxl288@case.edu
[*]*Department of EECS, Case Western Reserve University, Cleveland, OH 44106, USA*
[+]*College of Engineering, Virginia Commonwealth University, Richmond, VA 23284, USA*
[†]*Department of Computer and Information Sciences, Towson University, Towson, MD 21252, USA*

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Community detection is an effective approach to unveil social dynamics among individuals in social networks. In the literature, quite a few algorithms have been proposed to conduct community detection by exploiting the topology of social networks and the attributes of social actors. In practice, community detection is usually conducted by third parties like advertisement companies, hospitals, with access to social networks for different purposes, which can easily lead to privacy breaches. In this paper, we investigate community detection in social networks aiming to protect the privacy of both the network topologies and the users' attributes. In particular, we propose a new scheme called differentially private community detection (DPCD). DPCD detects communities in social networks via a probabilistic generative model, which can be decomposed into subproblems solved by individual users. The private social relationships and attributes of each user are protected by objective perturbation with differential privacy guarantees. Through both theoretical analysis and experimental validation using synthetic and real world social networks, we demonstrate that the proposed DPCD scheme detects social communities under modest privacy budget.

Keywords: social networks, community detection, differential privacy, objective perturbation

1. Introduction

Social media has been flourishing over the past few decades, and online social networks have become a global game changer for individuals to build their connections and for businesses to reach their potential customers. In an online social network, each entity, such as a Youtube subscriber or a Tweeter user, is represented by a node, while the relationship between a pair of nodes, such as subscription, retweet, or following, is characterized by an edge. Usually, social networks are self-organizing, emergent and complex, and many analysis tools are developed to discover their properties. Community detection serves as a fundamental tool to mine the complex topologies of social networks and understand the social interactions among individuals.

To identify the underlying community structures in social networks, researchers have developed many algorithms. Traditional community detection approaches usually focus on network topologies only (Plantié and Crampes (2013)). In fact, social actors in social networks, such as individuals or organizations, have their own characteristics or attributes, resulting in networks referred to as attributed networks (Huang et al. (2015)). In an attributed social network, network topologies and individuals' attributes jointly define communities (Yang et al. (2013)). For example, friends in a social circle have similar attributes such as interests, education background and interact frequently (Leskovec and Mcauley (2012)). Recently, quite a few community detection algorithms attempt to exploit both the topologies of social networks and the attributes of social actors (Li et al. (2018); He et al. (2017); Perozzi and Akoglu (2018); Yang et al. (2013, 2009)).

In practice, community detection in social networks is usually conducted by third-party agents (like data analyst, product provider or research organization) for different purposes. Thus, it can result in significant privacy breaches, compromising not only the relationships among users (through network topology) but also their sensitive personal information (through node attributes). For example, in early April 2018, it was reported that up to 87 million Facebook users' private personal information and their social relationships might have been shared with a political consulting company called Cambridge Analytica without their authorizations, which caused serious issues to Facebook. Obviously, it is critical to protect the privacy of social networks.

Some previous works have attempted to address the aforementioned privacy issue in social networks, which can be categorized as anonymization based, cryptographic techniques based, and differential privacy based schemes. Anonymization based schemes protect users' privacy by anonymizing their identities (Hay et al. (2007)). However, users' sensitive information may still be inferred (Narayanan and Shmatikov (2009)), and network topology is not protected. Cryptographic techniques based schemes usually incur expensive computations due to encryption and decryption operations (Dong et al. (2011); Jahid et al. (2011)), which makes them impractical for large-scale social networks. Previous differential privacy based schemes mostly try to protect the privacy of network topologies only, while ignoring users' sensitive information (Chen et al. (2014); Nguyen et al. (2015); Mülle et al. (2015); Nguyen et al. (2016); Su et al. (2016); Pinot et al. (2018)). For example, Nguyen et al. (2016) propose LouvainDP to apply input perturbation (Dwork and Roth (2014)) on Louvain method to conduct community detection on a noisy graph. They also propose ModDiv applying the exponential mechanism (Dwork and Roth (2014)) on community detection. Mülle et al. (2015) propose differentially private mechanism to flip the edges in graph when detecting communities. All these methods focus on the perturbation of graph structures only. Therefore, how to protect both the topology and users' attributes in social networks at the same time is still a challenging and open problem.

In this paper, we propose an effective community detection algorithm called differentially private community detection (DPCD) that can protect the privacy of network topologies and node attributes in attributed social networks. Our algorithm is built upon a generative probabilistic model that conducts community detection by solving a maximum log-likelihood problem, and the privacy is protected by applying objective perturbation (Chaudhuri et al. (2011); Chaudhuri and Monteleoni (2009)) with differential privacy guarantees. In particular, we decompose the maximum log-likelihood problem into convex subproblems, each of which deals with the social relationships and attributes of one particular user. To protect the private social relationships of each user, the objective function concerning his/her social relationships is perturbed by injected noise with designed probability distribution. To protect the privacy of users' attributes, each user is required to independently generate noise, whereas the summation of these noise satisfies the designed distribution.

We summarize the main contributions of this work as follows:

- We develop a differentially private community detection algorithm called DPCD that protects the privacy of both the social relationships and attributes of all users. Theoretical analysis shows that DPCD satisfies ϵ_G- and ϵ_X-differential privacy on network topology and node attributes, respectively.
- DPCD is effective with modest privacy budget and can accommodate both binary and continuous attributed social networks.
- Experiment results on both synthetic and real social networks demonstrate that the proposed DPCD achieves community detection results that are close to 3 non-private baselines and outperforms 7 other state-of-the-art privacy-preserving community detection schemes.

The rest of the paper is organized as follows. In Section 2.1, we first recall an existing community detection algorithm called CESNA, which is a building block of our DPCD mechanism, and then introduce the privacy model in Section 2.2. After that, we integrate differential privacy into CESNA to protect users' relationship and attributes in Section 3.1 and 3.2, respectively, and extend CESNA to accommodate social network with continuous attributed users in Section 3.3. In Section 4, we validate the proposed mechanism using both synthetic and real-world social networks. Finally, we conclude the paper in Section 5.

2. Problem Formulation

2.1. Community Detection in Attributed Social Networks

We consider a social network $G = (V, E)$, where V is the set of N nodes (users) and E is the set of edges (social connections). We assume that each node has K binary attributes, and that all the nodes in G can be clustered into C communities. To detect communities, we consider a generative probabilistic model called *CESNA* (Yang et al. (2013)), which is scalable and can detect communities even if some of them are overlapped. Specifically, we assume that the network adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$ and the binary node attributes $\mathbf{X} \in \{0,1\}^{N\times K}$ are generated from node community affiliation matrix $\mathbf{F} \in \mathcal{R}_+^{N\times C}$, i.e.,

$$\mathbf{A}_{ij} \sim \text{Bernoulli}(p_{ij}),\ p_{ij} = 1 - \exp(-\mathbf{f}_i\mathbf{f}_j^T), \tag{1}$$

$$\mathbf{x}_{ik} \sim \text{Bernoulli}(Q_{ik}),\ Q_{ik} = \frac{1}{1+\exp(-\mathbf{f}_i\mathbf{w}_k^T)}, \tag{2}$$

where $\mathbf{A}_{ij}$ being 1 means that there is an edge between node i and j, and 0 otherwise, p_{ij} is the probability of node i and j are connected in the social network, and $\mathbf{f}_i$ (quantify the affiliation of node i to all communities) is the i-th row of $\mathbf{F}$, $\mathbf{x}_{ik}$ is the binary value of the

k-th attribute of node i, Q_{ik} is the probability of $\mathbf{x}_{ik}$ equal to 1, and $\mathbf{w}_k \in \mathcal{R}^{C\times 1}$ is the logistic parameter for the k-th attribute.

Then the optimal $\hat{\mathbf{F}}$ and $\hat{\mathbf{W}}$ can be obtained by solving a maximum log-likelihood problem with l_1 regularization on $\mathbf{W}$ as:

$$\hat{\mathbf{F}}, \hat{\mathbf{W}} = \underset{\mathbf{F}\geq 0, \mathbf{W}}{\text{argmax}}\ \mathcal{L}_G + \mathcal{L}_\mathbf{X} - \lambda ||\mathbf{W}||_1, \tag{3}$$

where $\mathcal{L}_G = \sum_{(i,j)\in E} \log(1 - e^{-\mathbf{f}_i \mathbf{f}_j^T}) - \sum_{(i,j)\notin E} \mathbf{f}_i \mathbf{f}_j^T$ and $\mathcal{L}_\mathbf{X} = \sum_{i,k}(x_{ik}\log Q_{ik} + (1 - x_{ik})\log(1 - Q_{ik}))$ are the log-likelihood of the network topology and node attributes, respectively, and λ is the regularization hyper-parameter. To solve (3), we adopt the block coordinate ascent approach. The basic idea is to first decompose (3) into a set of convex subproblems that are easily solvable at each user, and then update $\mathbf{F}$ and $\mathbf{W}$ iteratively. Specifically, by fixing $\mathbf{W}$ and $\mathbf{F}$ but $\mathbf{f}_i$, we have the subproblems:

$$\hat{\mathbf{f}}_i = \underset{\mathbf{f}_i \geq 0}{\text{argmax}}\ \mathcal{L}_G(\mathbf{f}_i) + \mathcal{L}_\mathbf{X}(\mathbf{f}_i), i \in \{1, 2, \cdots N\}. \tag{4}$$

Similarly, we fix the updated $\mathbf{F}$ and obtain $\mathbf{W}$ by solving the following subproblems:

$$\hat{\mathbf{w}}_k = \text{argmax}\ \ \mathcal{L}_\mathbf{X}(\mathbf{w}_k) - \lambda ||\mathbf{w}_k||_1, k \in \{1, 2, \cdots K\}. \tag{5}$$

We apply gradient ascent to (4) and (5) to update $\mathbf{f}_i$ ($i \in \{1, 2, \cdots N\}$) and $\mathbf{w}_k$ ($k \in \{1, 2, \cdots K\}$) as follows:

$$\mathbf{f}_i^{new} \leftarrow max(0, \mathbf{f}_i^{old} + \alpha(\frac{\partial \mathcal{L}_G(\mathbf{f}_i)}{\partial \mathbf{f}_i} + \frac{\partial \mathcal{L}_\mathbf{X}(\mathbf{f}_i)}{\partial \mathbf{f}_i})) \tag{6}$$

$$\mathbf{w}_k^{new} \leftarrow \mathbf{w}_k^{old} + \alpha(\frac{\partial \mathcal{L}_\mathbf{X}(\mathbf{w}_k)}{\partial \mathbf{w}_k} - \lambda \text{Sign}(\mathbf{w}_k^{old})) \tag{7}$$

where the partial derivatives are calculated as:

$$\frac{\partial \mathcal{L}_G(\mathbf{f}_i)}{\partial \mathbf{f}_i} = \sum_{j\in\mathcal{N}(i)} \frac{exp(-\mathbf{f}_i\mathbf{f}_j^T)}{1 - exp(-\mathbf{f}_i\mathbf{f}_j^T)}\mathbf{f}_j - \sum_{j\notin\mathcal{N}(i)} \mathbf{f}_j, \tag{8}$$

$$\frac{\partial \mathcal{L}_\mathbf{X}(\mathbf{f}_i)}{\partial \mathbf{f}_i} = \sum_{k=1}^{K}(x_{ik} - Q_{ik})\mathbf{w}_k^T, \tag{9}$$

$$\frac{\partial \mathcal{L}_\mathbf{X}(\mathbf{w}_k)}{\partial \mathbf{w}_k} = \sum_{i\in\mathcal{N}}(x_{ik} - Q_{ik})\mathbf{f}_i^T. \tag{10}$$

Here, $\mathcal{N}(i) = \{j|(i,j) \in E\}$ represents the set of neighboring nodes of node i. After the update process terminates, we obtain $\hat{\mathbf{F}}$ and $\hat{\mathbf{W}}$. Then we assign node i to community c if $\hat{\mathbf{F}_{(ic)}}$ is larger than a predefined threshold δ.

2.2. Privacy Model

We consider differential privacy (Dwork and Roth (2014)) as the privacy model, because it offers provable guarantees of privacy without making assumptions about an adversary's prior knowledge. By injecting random noises into dataset, differential privacy mechanisms can prevent the adversary from determining if any individual user is in the dataset. Roughly speaking, differential privacy bounds the ratio between the probabilities of a randomized algorithm returning identical outcomes on two neighboring datasets which differ in only one record. To protect the privacy of network topology and node attributes, we define ϵ-differential privacy, neighboring graph, and edge-differential privacy in the following:

Definition 1 *ϵ-differential privacy (Dwork and Roth (2014)). A randomized algorithm $\mathcal{M}$ with domain $\mathcal{D}$ satisfies ϵ-differential privacy if for any two neighboring datasets $d, d' \in \mathcal{D}$, and for all $\mathcal{S} \subseteq \mathrm{Range}(\mathcal{M})$ it holds that $\Pr[\mathcal{M}(\mathrm{d}) \in \mathcal{S}] \leq \mathrm{e}^{\epsilon}\Pr[\mathcal{M}(\mathrm{d}') \in \mathcal{S}]$.*

Definition 1 has been adopted to protect private data by adding calibrated random noise. It can be interpreted as indistinguishability of two neighboring datasets, i.e., d, d', which is measured in a probabilistic manner given in Definition 1, with ϵ being a small positive constant. Thus, it ensures the probability of a statistical query producing a nearly same result whenever it is conducted on the first or second dataset.

Definition 2 *Neighboring graphs (Hay et al. (2009)). Given a graph G, a neighboring graph G' can be produced by either adding/removing an edge in E, or by adding/removing an isolated node in V. Mathematically, it is $|V \oplus V'| + |E \oplus E'| = 1$, where $\oplus$ stands for the symmetric difference.*

Note that in this paper we study community detection, where the output is the assignment of nodes to communities. Thus, we consider neighboring graphs that are produced by adding/removing of an edge.

Definition 3 *Edge-differential privacy (Hay et al. (2009)). The adaption of differential privacy to graphs is called edge-differential privacy.*

An edge-differentially private algorithm protects any individual edge in a network from disclosure.

3. A Differentially Private Community Detection Algorithm

In this section, we elaborate on the design of a differentially private community detection algorithm in social networks called DPCD, which performs objective perturbation under differential privacy.

3.1. Protection on Network Topology

In privacy-preserving community detection, one of the challenges is to enable a third-party agent to conduct community detection while not compromising users' sensitive social relationships, i.e., the social network topology. To protect the social relationships of node i, we perturb the objective function in (4) so that the solution of $\mathbf{f}_i$ is obfuscated. Particularly, we rewrite (4) as:

$$\hat{\mathbf{f}}_i = \underset{\mathbf{f}_i \geq 0}{\mathrm{argmax}}\ \mathcal{L}_{\mathbf{f}}(\mathbf{f}_i) = \underset{\mathbf{f}_i \geq 0}{\mathrm{argmax}}\ \mathcal{L}_G(\mathbf{f}_i) + \mathcal{L}_{\mathbf{X}}(\mathbf{f}_i) + \mathbf{f}_i\mathbf{n}_i^T, \tag{11}$$

where $\mathbf{n}_i^T \in \mathcal{R}^{C\times 1}$ is the injected noise vector. Then we obtain the following partial derivative:

$$\frac{\partial \mathcal{L}_{\mathbf{f}}(\mathbf{f}_i)}{\partial \mathbf{f}_i} = \frac{\partial \mathcal{L}_G(\mathbf{f}_i)}{\partial \mathbf{f}_i} + \frac{\partial \mathcal{L}_{\mathbf{X}}(\mathbf{f}_i)}{\partial \mathbf{f}_i} + \mathbf{n}_i. \tag{12}$$

In order to protect the social relationships of node i, the partial derivative is calculated at node i so that the private social connection information, i.e., $\mathcal{N}(i)$, is kept locally and protected from the third-party. Moreover, the injected noise needs to guarantee that the derived $\mathbf{f}_i$ satisfies edge-differential privacy. Notice that, although each element of $\mathbf{f}_i$ is positive, its norm should have an upper bound, since (3) is a variation of nonnegative matrix factorization. We denote $||\mathbf{f}_i||_2 \le \Delta, \forall i \in V$, and set $\Delta = 1$ according to (Yang and Leskovec (2013)). Besides, if node i and j do not belong to any common community, then $\mathbf{f}_i\mathbf{f}_j^T = 0$, which makes $p_{ij} = 0$. To account for the possibility that node i and j are actually connected even if they do not share common communities, we let $p_{ij} \ge \rho$, where the lower bound ρ can be set as the network density in practice.

Consequently, to achieve edge-differential privacy in the course of updating $\mathbf{f}_i$, we require user i to generate $\mathbf{n}_i$ whose element is distributed as $Lap(\frac{\Delta\sqrt{C}}{\rho\epsilon_G})$, where ϵ_G is the privacy budget in network topology protection. Theorem 1 demonstrates that under this condition the third-party cannot infer whether user i is connected to any other user j or not in a given social network. Therefore, the privacy of the social network topology is protected.

Theorem 1 *If each element in $\mathbf{n}_i$ is independently and randomly selected from $Lap(\frac{\Delta\sqrt{C}}{\rho\epsilon_G})$, then the proposed DPCD is ϵ_G-edge-differentially private.*

Proof Suppose G_1 and G_2 are neighboring graphs differing by one edge as in Definition 2. When Algorithm 1 converges, we have $\frac{\partial \mathcal{L}_{\mathbf{f}}(\mathbf{f}_i|G_1)}{\partial \mathbf{f}_i} = \frac{\partial \mathcal{L}_{\mathbf{f}}(\mathbf{f}_i|G_2)}{\partial \mathbf{f}_i} = 0$, i.e.,

$$\begin{aligned} &\sum_{j\in\mathcal{N}_1(i)} \frac{exp(-\mathbf{f}_i\mathbf{f}_j^T)}{1-exp(-\mathbf{f}_i\mathbf{f}_j^T)}\mathbf{f}_j - \sum_{j\notin\mathcal{N}_1(i)} \mathbf{f}_j + \sum_{k=1}^{K}(x_{ik}-Q_{ik})\mathbf{w}_k^T + \mathbf{n}_i^T \\ =&\sum_{j\in\mathcal{N}_2(i)} \frac{exp(-\mathbf{f}_i\mathbf{f}_j^T)}{1-exp(-\mathbf{f}_i\mathbf{f}_j^T)}\mathbf{f}_j - \sum_{j\notin\mathcal{N}_2(i)} \mathbf{f}_j + \sum_{k=1}^{K}(x_{ik}-Q_{ik})\mathbf{w}_k^T + \mathbf{n'}_i^T \end{aligned} \tag{13}$$

If node i is not associated with the differing edge between G_1 and G_2, we have $||\mathbf{n}_i||_2 = ||\mathbf{n}'_i||_2$. If the differing edge between G_1 and G_2 involves user i and another user k due to the adding or removing the edge between them, then we have $||\mathbf{n}_i - \mathbf{n}'_i||_2 = \frac{1}{p_{ik}}||\mathbf{f}_k||_2 \le \frac{\Delta}{\rho}$, where the inequality follows that the norm of each affiliation vector is upper bounded by Δ and the connection probability between any pair of nodes is practically lower bounded by ρ. Then for any pair of neighboring G_1 and G_2, we have:

$$\begin{aligned} \frac{\Pr(\hat{\mathbf{f}}^* = \hat{\mathbf{f}}_i|G_1)}{\Pr(\hat{\mathbf{f}}^* = \hat{\mathbf{f}}_i|G_2)} &= \frac{\prod_{c\in\{1,2,\cdots,C\}}\Pr(\mathbf{n}_{i_c})}{\prod_{c\in\{1,2,\cdots,C\}}\Pr(\mathbf{n}'_{i_c})} = \exp(\frac{\rho\epsilon_G}{\Delta\sqrt{C}}\sum_c(|\mathbf{n}'_{i_c}| - |\mathbf{n}_{i_c}|)) \\ &\le \exp(\frac{\rho\epsilon_G}{\Delta\sqrt{C}}||\mathbf{n}'_i - \mathbf{n}_i||_1) \le \exp(\frac{\rho\epsilon_G}{\Delta\sqrt{C}}\sqrt{C}||\mathbf{n}_i - \mathbf{n}'_i||_2) \le e^{\epsilon_G}. \end{aligned}$$

Therefore, we conclude the proof. ■

Moreover, the computation of (9) involves only $\mathbf{x}_i$ and $\mathbf{W}$, and does not require any information related to network topology. The logistic parameter $\mathbf{W}$ can be shared in the social network, because it does not contain any users' information. Besides, each user also keeps his attribute vector, i.e., $\mathbf{x}_i$, to himself, thus $\frac{\partial \mathcal{L}_{\mathbf{X}}(\mathbf{f}_i)}{\partial \mathbf{f}_i}$ can be computed locally by each user without privacy breach. The steps to update $\mathbf{F}$ with differential privacy guarantee are summarized in the first for-loop in Algorithm 1.

Algorithm 1 Differentially private community detection (DPCD) algorithm

```
Input: Privacy budget: $\epsilon_G$, $\epsilon_X$. Each user $i$ holds his private attributes, i.e., $\mathbf{x}_i$ and private social connections, i.e., $\mathcal{N}_i$.
Output: $\hat{\mathbf{F}}$, $\hat{\mathbf{W}}$
while the value of objective function in (3) still increases do
    for user $i \in \{1, 2, \cdots, N\}$ do
        query $\mathbf{f}_j$ from other users
        generate $\mathbf{n}_i \sim Lap(\frac{2\Delta\sqrt{C}}{\rho\epsilon_G})$
        compute $\frac{\partial \mathcal{L}_{\mathbf{f}}(\mathbf{f}_i)}{\partial \mathbf{f}_i}$ in (12)
        $\mathbf{f}_i^{new} \leftarrow max(0, \mathbf{f}_i^{old} + \alpha\frac{\partial \mathcal{L}_{\mathbf{f}}(\mathbf{f}_i)}{\partial \mathbf{f}_i})$
    end
    for $k \in \{1, 2, \cdots, C\}$ do
        agent sends $\mathbf{h}_k \sim Exp(1)$ to all users
        for user $i \in \{1, 2, \cdots, N\}$ do
            generate $\mathbf{u}_i \sim \mathcal{N}(0, 1/N)$
            compute $\mathbf{n}_{k_i} = \frac{\Delta\sqrt{C+1}}{\epsilon_{\mathbf{X}}}\sqrt{2\mathbf{h}_k} \circ \mathbf{u}_i$
            send $(x_{ik} - Q_{ik})\mathbf{f}_i^T + \mathbf{n}_{k_i}$ to the agent
        end
        agent calculates cumulative sum of $(x_{ik} - Q_{ik})\mathbf{f}_i^T + \mathbf{n}_{k_i}$ and subtracts $\lambda\text{Sign}(\mathbf{w}_k)$ to get $\frac{\partial \mathcal{L}_{\mathbf{w}}(\mathbf{w}_k)}{\partial \mathbf{w}_k}$ in (15)
        $\mathbf{w}_k^{new} \leftarrow \mathbf{w}_k^{old} + \alpha\frac{\partial \mathcal{L}_{\mathbf{w}}(\mathbf{w}_k)}{\partial \mathbf{w}_k}$
    end
end
Return $\hat{\mathbf{F}}$, $\hat{\mathbf{W}}$
```

3.2. Protection on Users' Attributes

Users' attributes in social networks contain their sensitive information, such as interests, education background, relationship and sexual orientation, etc. Thus, protecting the privacy of users' attributes during community detection is of great importance.

Recall that $\mathbf{W}$ is shared in the social network, and $\mathbf{w}_k$ is updated by the agent. The partial derivative of $\mathcal{L}_{\mathbf{X}}$ w.r.t. $\mathbf{w}_k$ in (10) requires each user i to send $(x_{ik} - Q_{ik})\mathbf{f}_i^T$ to the agent. Since $\mathbf{f}_i$ is known to the agent and x_{ik} is either 0 or 1, the agent can easily recover x_{ik} based on $x_{ik} - Q_{ik}$. Particularly, if $x_{ik} - Q_{ik} > 0$, then $x_{ik} = 1$, otherwise $x_{ik} = 0$.

To prevent the information leakage of users' attributes, we perturb the objective function concerning all node attributes in (5) as follows:

$$\hat{\mathbf{w}}_k = \underset{\mathbf{w}_k}{\text{argmax}}\ \mathcal{L}_{\mathbf{w}}(\mathbf{w}_k) = \underset{\mathbf{w}_k}{\text{argmax}}\ \mathcal{L}_{\mathbf{X}}(\mathbf{w}_k) - \lambda||\mathbf{w}_k||_1 + \mathbf{n}_k^T \mathbf{w}_k, \tag{14}$$

where $\mathbf{n}_k \in \mathcal{R}^{1\times C}$ is the injected noise.

Then, we have

$$\frac{\partial \mathcal{L}_{\mathbf{w}}(\mathbf{w}_k)}{\partial \mathbf{w}_k} = \frac{\partial \mathcal{L}_{\mathbf{X}}(\mathbf{w}_k)}{\partial \mathbf{w}_k} - \lambda \ \text{Sign}(\mathbf{w}_k) + \mathbf{n}_k. \tag{15}$$

We further decompose $\mathbf{n}_k = \sum_i \mathbf{n}_{k_i}$, where $\mathbf{n}_{k_i}$ is the noise vector generated by each user. In other words, each user adds $\mathbf{n}_{k_i}$ to $(x_{ik} - Q_{ik})\mathbf{f}_i^T$ before submitting it to the agent.

How to generate the noise vector at each user is very critical. We propose the following procedure to help users generate noises: first, the agent generates $\mathbf{h}_k \in \mathcal{R}^{C\times 1}$, the elements of which are i.i.d. sampled from $Exp(1)$, and send it to all users. Then, each user generates $\mathbf{u}_i \in \mathcal{R}^{C\times 1}$, the elements of which are i.i.d. sampled from $\mathcal{N}(0, 1/N)$. At last, each user computes its noise vector $\mathbf{n}_{k_i} = \frac{\Delta\sqrt{C}}{\epsilon_{\mathbf{X}}}\sqrt{2\mathbf{h}_k} \circ \mathbf{u}_i$, where $\circ$ is the Hadamard product, and ϵ_X is the privacy budget in attributes protection. This procedure is summarized in the second for-loop in Algorithm 1.

The Laplace noise generation process is motivated by the following lemma.

Lemma 2 *For random numbers $h \sim Exp(1)$, $u \sim \mathcal{N}(0, 1)$, and $\lambda > 0$, we have $\lambda\sqrt{2h}u \sim$ Lap(λ) Kotz et al. (2012).*

Proposition 1 *Let $\mathbf{n}_k = \sum_i \mathbf{n}_{k_i}$. Then, each element of $\mathbf{n}_k$ follows the distribution of Lap$(\frac{\Delta\sqrt{C}}{\epsilon_{\mathbf{X}}})$.*

Proof

$$\mathbf{n}_k = \sum_i \mathbf{n}_{k_i} = \frac{\Delta\sqrt{C}}{\epsilon_{\mathbf{X}}} \sum_i \sqrt{2\mathbf{h}_k} \circ \mathbf{u}_i \overset{*}{=} \frac{\Delta\sqrt{C}}{\epsilon_{\mathbf{X}}} \sqrt{2\mathbf{h}_i} \circ \mathbf{u},$$

where each element in $\mathbf{u} = \sum_i \mathbf{u}_i$ is distributed as $\mathcal{N}(0, 1)$ and $*$ follows that the summation of Gaussian distributed variables is still distributed as Gaussian. Thus, according to Lemma 2, each element of $\mathbf{n}_k$ is distributed as $Lap(\frac{\Delta\sqrt{C}}{\epsilon_{\mathbf{X}}})$. ■

Next, we prove that any third-party agent cannot infer the value of the attributes of an arbitrary user, and the privacy of node attributes is protected with privacy budget $\epsilon_{\mathbf{X}}$.

Theorem 3 *If each element of $\mathbf{n}_k$ is independently and randomly selected from Lap$(\frac{\Delta\sqrt{C}}{\epsilon_{\mathbf{X}}})$, then the proposed DPCD is $\epsilon_{\mathbf{X}}$-differentially private on binary users' attributes.*

Proof Given two attributed social networks whose neighboring attribute matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ are different in one record, e.g., x_{uv} for $\mathbf{X}_1$ and x'_{uv} for $\mathbf{X}_2$. After the convergence of Algorithm 1, we have $\frac{\partial \mathcal{L}_{\mathbf{w}}(\mathbf{w}_k|\mathbf{X}_1)}{\partial \mathbf{w}_k} = \frac{\partial \mathcal{L}_{\mathbf{w}}(\mathbf{w}_k|\mathbf{X}_2)}{\partial \mathbf{w}_k} = 0$. Thereby,

$$\sum_i (x_{ik} - Q_{ik})\mathbf{f}_i^T - \lambda \ \text{Sign}(\mathbf{w}_k) + \mathbf{n}_k = \sum_i (x_{ik} - Q_{ik})\mathbf{f}_i^T - \lambda \ \text{Sign}(\mathbf{w}_k) + \mathbf{n}'_k.$$

If $v \neq k$, then we have $||\mathbf{n}_k - \mathbf{n}'_k||_2 = 0$. If $v = k$, then $||\mathbf{n}_k - \mathbf{n}'_k||_2 = ||(x_{uv} - x'_{uv})\mathbf{f}_i^T||_2 \leq \Delta$ since the attributes are binary. Thus, for the two neighboring attribute matrix $\mathbf{X}_1$ and $\mathbf{X}_2$ we have:

$$\frac{\Pr(\hat{\mathbf{w}_k^*} = \hat{\mathbf{w}}_k|\mathbf{X}_1)}{\Pr(\hat{\mathbf{w}_k^*} = \hat{\mathbf{w}}_k|\mathbf{X}_2)} = \frac{\prod_{c\in\{1,2,\cdots,C\}} \Pr(\mathbf{n}_{k_c})}{\prod_{c\in\{1,2,\cdots,C\}} \Pr(\mathbf{n}'_{k_c})} = \exp(\frac{\epsilon_{\mathbf{X}} \sum_{c=1}^{C}(|\mathbf{n}'_{k_c}| - |\mathbf{n}_{k_c}|)}{\Delta\sqrt{C}})$$
$$\leq \exp(\frac{\epsilon_{\mathbf{X}}\sqrt{C}||\mathbf{n}'_k - \mathbf{n}_k||_2}{\Delta\sqrt{C}}) \leq e^{\epsilon_{\mathbf{X}}}.$$

Thus, we conclude the proof. ■

Up to now, we have described the differentially private community detection (DPCD) algorithm for protecting network topology and user attributes in binary attributed social networks. We summarize the DPCD algorithm in Algorithm 1, where each user holds its own attributes vector, i.e., $\mathbf{x}_i$, and social relationships, i.e., $\mathcal{N}(i)$. The inputs to DPCD are the privacy budget (ϵ_G and ϵ_X), and the outputs are differentially private affiliation matrix $\hat{\mathbf{F}}$, and logistic parameter $\hat{\mathbf{W}}$. Since we employ block-coordinate ascent method, then the computational complexity for all nodes at one iteration in Algorithm 1 is $O(|E| + NK)$, which means the proposed DPCD is efficient and scalable.

3.3. Extension to Social Networks with Continuous Attributes

In real social networks, besides binary attributes, users can also have continuous attributes, e.g., ratings in recommendation systems or posts in social platforms. Therefore, we extend the proposed DPCD to accommodate nodes with continuous attributes which are normalized to 1.

First, we consider a multivariate regression model for each user's attributes as $\mathbf{x}_i = \mathbf{f}_i\mathbf{W} + \mathbf{b}$, where $\mathbf{W} \in \mathcal{R}^{C\times K}$ is the weight for the multivariate regression, and $\mathbf{b} \in \mathcal{R}^{1\times K}$ is the bias vector. Usually, for a multivariate regression model, its residual of the fitting result, i.e., $\mathbf{e}_i = \mathbf{f}_i\mathbf{W} + \mathbf{b} - \mathbf{x}_i$, is assumed to be a multivariate Gaussian error vector (Var (1998)). This means that $\mathbf{e}_i$ is i.i.d. sampled from distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Hence, according to (Tabachnick et al. (2007)) the likelihood of the node attributes is:

$$\Pr(\text{vec}(\mathbf{X})|\mathbf{F}, \mathbf{W}) \sim \mathcal{N}([\mathbf{W}^T \otimes \mathbf{I}_{N\times N}]\text{vec}(\mathbf{F}), \mathbf{\Sigma} \otimes \mathbf{I}_{N\times N}),$$

where vec$(\cdot)$ denotes the vectorization operator that stacks all columns of a matrix into a vector. As a result, we can rewrite (3) as:

$$\hat{\mathbf{F}}, \hat{\mathbf{W}} = \underset{\mathbf{F}\geq 0, \mathbf{W}}{\text{argmax}}\ \mathcal{L}_G + \mathcal{L}_{\mathbf{X}} - \lambda||\mathbf{W}||_1$$
$$= \underset{\mathbf{F}\geq 0, \mathbf{W}}{\text{argmax}} \sum_{(i,j)\in E} \log(1 - e^{-\mathbf{f}_i\mathbf{f}_j^T}) - \sum_{(i,j)\notin E} \mathbf{f}_i\mathbf{f}_j^T - ||\mathbf{X} - \mathbf{F}\mathbf{W}||_F^2 - \lambda||\mathbf{W}||_1$$

Compared to (3), we can see that $\mathcal{L}_{\mathbf{X}}$ is replaced by $-||\mathbf{X} - \mathbf{F}\mathbf{W}||_F^2$. This is because the least-square regression corresponds to finding the maximum likelihood estimation of parameters in a Gaussian distribution. Then we can obtain the partial derivatives of $\mathcal{L}_{\mathbf{X}}$ w.r.t. $\mathbf{f}_i$ and $\mathbf{w}_k$ as follows:

$$\frac{\partial \mathcal{L}_{\mathbf{X}}(\mathbf{f}_i)}{\partial \mathbf{f}_i} = -2(\mathbf{f}_i \mathbf{W} - \mathbf{x}_i)\mathbf{W}^{\mathbf{T}}, \tag{16}$$

$$\frac{\partial \mathcal{L}_{\mathbf{X}}(\mathbf{w}_k)}{\partial \mathbf{w}_k} = -2\sum_i \mathbf{f}_i^T(\mathbf{f}_i \mathbf{w}_k - x_{ik}). \tag{17}$$

Note that (16) can be computed locally by individual users. Therefore, similar to that in Section 3.1, we can add perturbation noises to (4) to protect the privacy of network topology using the same procedure in Algorithm 1. Thus we have the following theorem.

Theorem 4 *If each element in* $\mathbf{n}_i$ *is independently and randomly selected from Lap*$(\frac{\Delta\sqrt{C}}{\rho\epsilon_{G'}})$*, where* $\epsilon_{G'}$ *is the privacy budget, then the proposed DPCD is* $\epsilon_{G'}$*-edge-differentially private on continuous attributed social networks.*

Proof The proof follows a similar approach to that for Theorem 1. Thus, we omit the proof here. ■

Moreover, similar to that in Section 3.2, we add perturbation noises to (5) so as to protect the privacy of user' attributes, i.e.,

$$\hat{\mathbf{w}}_k = \underset{\mathbf{w}_k}{\operatorname{argmax}}\ \mathcal{L}'_{\mathbf{w}}(\mathbf{w}_k) = \underset{\mathbf{w}_k}{\operatorname{argmax}}\ \mathcal{L}_{\mathbf{X}}(\mathbf{w}_k) - \lambda||\mathbf{w}_k||_1 + \mathbf{n}_k^T \mathbf{w}_k, \tag{18}$$

$$\frac{\partial \mathcal{L}'_{\mathbf{w}}(\mathbf{w}_k)}{\partial \mathbf{w}_k} = -2\sum_i \mathbf{f}_i^T(\mathbf{f}_i \mathbf{w}_k - x_{ik}) - \lambda\ \mathrm{Sign}(\mathbf{w}_k) + \sum_i \mathbf{n}_{k_i}. \tag{19}$$

Subsequently, we have the theorem below about the differential privacy of the continuous user attributes.

Theorem 5 *If each element of* $\mathbf{n}_k = \sum_i \mathbf{n}_{k_i}$ *is independently and randomly selected from Lap*$(\frac{2\Delta\sqrt{C}}{\epsilon_{\mathbf{X}'}})$*, where* $\epsilon_{\mathbf{X}'}$ *is the privacy budget, then the proposed DPCD scheme is* $\epsilon_{\mathbf{X}'}$*-differentially private on continuous users' attributes.*

Proof The proof follows a similar approach to that for Theorem 3 and is omitted here. ■

4. Experiments

In this section, we evaluate the performance of the proposed DPCD using both synthetic and real attributed social networks. In all experiments, 100 trials are carried out and the averages are reported. In what follows, we present the dataset, the evaluation metric, the effectiveness of the proposed algorithm, and the effect of the privacy budgets, respectively.

4.1. Dataset Description

We synthesize two social networks using two different approaches. The first social network, referred to as Gen, is generated by the CESNA model (Yang et al. (2013)). Specifically, we first initialize $\mathbf{F} \in \mathcal{R}_+^{N \times C}$, $\mathbf{F}_{(ic)} \in [0, 1]$, keep its element only if it is larger than 0.95 quantile of $\mathbf{F}$, and then generate the network topology ($\mathbf{A}$) and node attributes ($\mathbf{X}$) according to (1) and (2), respectively. To generate the ground truth communities, we assign node i to community c, if $\mathbf{F}_{(ic)}$ is kept. The second synthetic social network, referred to as SBM, is generated by employing the stochastic block model (Airoldi et al. (2008)). Specifically, this model assumes that the network contains C blocks. It establishes an edge inside a block with probability p_{intra}, and between two blocks with probability p_{inter}. The attributes of nodes in a community c are generated by drawing i.i.d. samples from Bernoulli(p_c). The C blocks are then considered to be the ground truth communities during evaluation.

On the other hand, we also collect data from real-world social networks. For a social network with binary attributes, we consider a published Facebook dataset, which is a set of 10 ego-networks (Yang and Leskovec (2013)). The attributes come from the profile of Facebook users, such as education, gender, job description. Each ego-network contains manually labeled social circles, while merging the 10 ego-networks gives a total of 193 social circles. In the experiments, we detect both 10 ego-networks and 193 social circles, and call them FB1 and FB2, respectively. We use the ego-circles, each of which is a cluster of connections between a user and his friends, as ground-truth communities in Facebook dataset for two reasons. First, it is commonly adopted by many other community detection works, such as CESNA (Yang and Leskovec (2013)) and CDE (Li et al. (2018)). To make fair comparisons, we choose the same set of ground-truth communities. Second, the available Facebook dataset is incomplete, since the data was collected from survey participants using an App and the network is built by merging the ego-circles. In this case, using the original ego-circles as ground-truth is straightforward and intuitive.

For social networks with continuous attributes, we build two retweet networks from crawled tweets posted on August 8th 2016, 2nd day of Rio Summer Olympics. The node attributes are vectors converted from the comments of the original tweets by the sentence embedding technique (Pennington et al. (2014)). We define the ground truth communities to be topics (shared hashtags in tweets). That is, if two tweets share the same set of hashtags, they belong to the same community. In the experiments, we define two different hashtag sets [1] of size 2 and 3, and the resulting retweet networks are referred as *ReT1* and *ReT2*, respectively. The statistics of all the aforementioned social networks are summarized in the left panel of Table 1.

4.2. Evaluation Metrics

To evaluate the performance of our proposed algorithm, we adopt the average F_1 score (Yang and Leskovec (2013)) as the evaluation metric. Given the detected communities, each community is matched with the most similar one in the ground-truth communities, and the F_1 score of the two matched sets c_1 and c_2 are computed as $F_1(c_1, c_2) = 2\frac{prec(c_1,c_2) \times recall(c_1,c_2)}{prec(c_1,c_2)+recall(c_1,c_2)}$,

[1] For example, for hashtag sets of size 2, we choose {#Rio2016, #TeamUSA}, {#Rio2016, #swimming}, {#Rio2016, #weightlifting}, {#Rio2016, #waterpolo}, {#Rio2016, #Basketball}, {#Rio2016, #Hockey} to define 6 communities as ground truth.

Dataset	N	E	C	K	D	CESNA	SCI	CDE	DPCD
Gen	1,000	11,450	30	15	0.023	0.519	0.399	0.412	0.336
SBM	3,000	62,717	30	20	0.014	0.806	0.639	0.734	0.694
FB1	4,039	88,234	10	10	0.011	0.594	0.504	0.650	0.412
FB2	4,039	88,234	193	10	0.011	0.365	0.077	0.360	0.311
ReT1	15,743	123,921	6	100	0.001	0.300	0.179	0.229	0.239
ReT2	172,821	11,677,129	47	100	0.001	0.561	0.183	0.499	0.452

Table 1: Left panel: statistics of social networks. N: node number, E: edge number, C: community number, K: attribute number, D: network density. Right panel: average F_1 score of detected communities. CESNA, SCI and CDE are the community detection algorithms without taking privacy into consideration; DPCD is our proposed differentially private algorithm.

where $prec(c_1, c_2) = \frac{|c_1 \cap c_2|}{|c_1|}$, and $recall(c_1, c_2) = \frac{|c_1 \cap c_2|}{|c_2|}$. Furthermore, we can have the average F_1 score of two sets of communities C and C^* as $\bar{F}_1(C, C^*) = \frac{1}{2|C|}\sum_{c_i \in C} F_1(c_i, C^*) + \frac{1}{2|C^*|}\sum_{c_i^* \in C^*} F_1(c_i^*, C)$, where $F_1(c_i, C^*) = \max_{c_j \in C^*} F_1(c_i, c_j)$.

4.3. The Effectiveness of the Proposed Algorithm

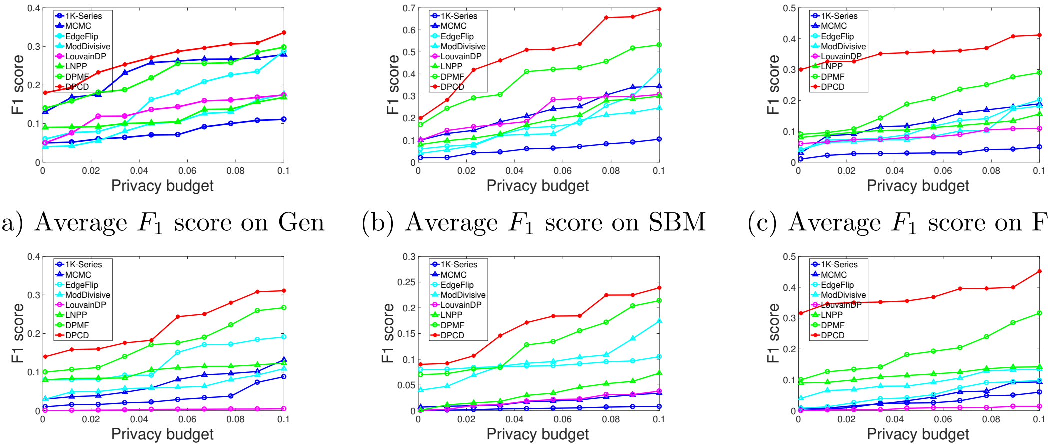
To show the effectiveness of DPCD, we compare the average F_1 score achieved by DPCD with that of other 3 state-of-the-art community detection algorithms without the consideration of the privacy protection. These algorithms are CESNA (Yang et al. (2013)), SCI (Wang et al. (2016)) and CDE (Li et al. (2018)). In the experiments, we set the threshold $\delta = 0.01$, and the privacy budget on network topology and node attributes as $\epsilon_G = \epsilon_{\mathbf{X}} = 0.1$. We show the comparison results in the right panel of Table 1. From this table, we can see that our proposed algorithm can achieve the average F_1 score close to that of non-private baselines. For example, for the ReT1 dataset, the F_1 score are $0.239, 0300, 0.179$ and 0.229 for DPCD, CESNA, SCI and CDE, respectively. It means that our algorithm achieves high utility of community detection results even under limited privacy budget.

On the other hand, we also compare the proposed algorithm with other 7 state-of-the-art algorithms that protects edge privacy only. They are $1K$-Series (Wang and Wu (2013)), MCMC (Xiao et al. (2014)), EdgeFlip (Mülle et al. (2015)), LouvainDP, ModDiv (Nguyen et al. (2016)), LNPP (Wang et al. (2013)) and DPMF (Hua et al. (2015)). Among them $1K$-Series and MCMC are the best known algorithms for graph structure release under differential privacy, which can be applied first on the social networks before running any off the shelf community detection algorithms. LNPP is a Laplace based mechanism that calibrates Laplace noise on the eigenvalues and preserves the spectral decomposition of matrices. DPMF performs matrix factorization with differentially private guarantee, so that we can adopt a nonnegative matrix factorization approach to conduct community detection (Yang and Leskovec (2013)).

In the experiment, we run $1K$-Series, MCMC and LNPP followed by CESNA. Because, CESNA is known to be one of the best community detection algorithms considering both edge structure and node attributes, $1K$-Series, MCMC and LNPP can get better results when working with CESNA. Besides, CESNA serves as a building block of our mechanism and we are making fair comparison by letting $1K$-Series, MCMC and LNPP work with CESNA. We set the privacy budget $\epsilon_G = \epsilon_{\mathbf{X}} = 0.1$, and use the average F_1 score to measure their performance. The comparison results are summarized in Table 2. We can see that under the same privacy budget, our proposed algorithm outperforms the other algorithms on all datasets with significant improvements. The main reason is that DPCD uses objective perturbation, whereas the other methods use input perturbation and do not take users' attributes into account. In differential privacy, objective perturbation is superior to the previous state-of-the-art, such as input and output perturbation, in managing the inherent tradeoff between privacy and learning performance (Chaudhuri et al. (2011)).

Dataset	$1K$-Series	MCMC	EdgeFlip	ModDiv	LouvainDP	LNPP	DPMF	DPCD
Gen	0.111	0.279	0.288	0.176	0.174	0.168	0.298	**0.336**
SBM	0.104	0.344	0.415	0.246	0.307	0.300	0.532	**0.694**
FB1	0.049	0.189	0.202	0.182	0.109	0.156	0.371	**0.412**
FB2	0.088	0.132	0.191	0.109	0.002	0.123	0.267	**0.311**
ReT1	0.008	0.034	0.105	0.174	0.038	0.073	0.214	**0.239**
ReT2	0.060	0.093	0.098	0.134	0.014	0.142	0.316	**0.452**

Table 2: Average F_1 score of detected communities by algorithms with differential privacy.



(a) Average F_1 score on Gen (b) Average F_1 score on SBM (c) Average F_1 score on FB1

(d) Average F_1 score on FB2 (e) Average F_1 score on ReT1 (f) Average F_1 score on ReT2

Figure 1: Average F_1 score versus privacy budget on all social networks.

4.4. The Effect of the Privacy Budget

To show the impact of the privacy budget on the performance of the proposed DPCD algorithm, we vary the privacy budget from 10^{-3} to 10^{-1} and calculate the corresponding average F_1 scores. We compare our proposed algorithm with other algorithms with privacy protection (i.e., $1K$-Series, MCMC, EdgeFlip, ModDiv, LouvainDP, LNPP and DPMF). 100 trials are carried out for all algorithms and the averages are reported. The comparison results are shown in Figure 1. We find that our proposed algorithm can achieve the highest F_1 score on all the datasets, which demonstrates that DPCD has the best performance. For example, in FB1 and ReT2 social networks, we achieve the highest average F_1 score with the lowest privacy budget ($\epsilon = 10^{-3}$). Actually, for $1K$-Series, MCMC, EdgeFlip, ModDiv and LouvainDP to have decent community detection results, they usually require $\epsilon = 0.5 \ln(N)$ (Nguyen et al. (2016)), which makes their privacy budget much higher than DPCD. On the other hand, these baseline algorithms can only protect the privacy of social network topology, whereas DPCD offers protection on both the network topology and users attributes, which is one of our main contributions. Moreover, we can see that in Figure 1, the average F_1 score grows as the privacy budget increases. This is consistent with the theoretical analysis.

5. Conclusions

In this paper, we have studied the problem of protecting the sensitive information when community detection is conducted in attributed social networks. In particular, we propose the DPCD algorithm to protect the privacy of both network topology and node attributes. Specifically, the community detection problem is formulated as a maximum log-likelihood problem that is decomposed into a set of convex subproblems. To protect the privacy of social networks, we add carefully generated noises to the objective function of these subproblems with differential privacy guarantees. Particularly, we theoretically prove that DPCD can achieve ϵ_G-edge-differential privacy on network topology and ϵ_X-differential privacy on node attributes. We implement the proposed algorithm and some existing algorithms, and evaluate the performance by considering both synthetic and real-world social networks. The experiment results show that the proposed DPCD achieves significant performance improvement compared to existing privacy-preserving community detection schemes.

References

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep):1981–2014, 2008.

Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *Advances in neural information processing systems*, pages 289–296, 2009.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Rui Chen, Benjamin CM Fung, S Yu Philip, and Bipin C Desai. Correlated network data publication via differential privacy. *The VLDB Journal*, 23(4):653–676, 2014.

Wei Dong, Vacha Dave, Lili Qiu, and Yin Zhang. Secure friend discovery in mobile social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 1647–1655. IEEE, 2011.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. Anonymizing social networks. *Computer science department faculty publication series*, page 180, 2007.

Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 169–178. IEEE, 2009.

Dongxiao He, Zhiyong Feng, Di Jin, Xiaobao Wang, and Weixiong Zhang. Joint identification of network communities and semantics via integrative modeling of network topologies and node contents. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jingyu Hua, Chang Xia, and Sheng Zhong. Differentially private matrix factorization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1763–1770. AAAI Press, 2015. ISBN 978-1-57735-738-4. URL http://dl.acm.org/citation.cfm?id=2832415.2832494.

Xin Huang, Hong Cheng, and Jeffrey Xu Yu. Dense community detection in multi-valued attributed networks. *Information Sciences*, 314:77–99, 2015.

Sonia Jahid, Prateek Mittal, and Nikita Borisov. Easier: Encryption-based access control in social networks with efficient revocation. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, pages 411–415. ACM, 2011.

Samuel Kotz, Tomasz Kozubowski, and Krzystof Podgorski. *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance.* Springer Science & Business Media, 2012.

Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

Ye Li, Chaofeng Sha, Xin Huang, and Yanchun Zhang. Community detection in attributed graphs: An embedding approach. In *AAAI*, 2018.

Yvonne Mülle, Chris Clifton, and Klemens Böhm. Privacy-integrated graph clustering through differential privacy. In *EDBT/ICDT Workshops*, pages 247–254, 2015.

Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE, 2009.

Hiep H Nguyen, Abdessamad Imine, and Michaël Rusinowitch. Differentially private publication of social graphs at linear cost. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 596–599. IEEE, 2015.

Hiep H Nguyen, Abdessamad Imine, and Michaël Rusinowitch. Detecting communities under differential privacy. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, pages 83–93. ACM, 2016.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Bryan Perozzi and Leman Akoglu. Discovering communities and anomalies in attributed graphs: Interactive visual exploration and summarization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(2):24, 2018.

Rafael Pinot, Anne Morvan, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. Graph-based clustering under differential privacy. *arXiv preprint arXiv:1803.03831*, 2018.

Michel Plantié and Michel Crampes. Survey on social community detection. In *Social media retrieval*, pages 65–85. Springer, 2013.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pages 26–37. ACM, 2016.

Barbara G Tabachnick, Linda S Fidell, and Jodie B Ullman. *Using multivariate statistics*, volume 5. Pearson Boston, MA, 2007.

I Var. Multivariate data analysis. *vectors*, 8(2):125–136, 1998.

Xiao Wang, Di Jin, Xiaochun Cao, Liang Yang, and Weixiong Zhang. Semantic community identification in large attribute networks. In *AAAI*, pages 265–271, 2016.

Yue Wang and Xintao Wu. Preserving differential privacy in degree-correlation based graph generation. *Transactions on data privacy*, 6(2):127, 2013.

Yue Wang, Xintao Wu, and Leting Wu. Differential privacy preserving spectral graph analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 329–340. Springer, 2013.

Qian Xiao, Rui Chen, and Kian-Lee Tan. Differentially private network data release via structural inference. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 911–920. ACM, 2014.

Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.

Jaewon Yang, Julian McAuley, and Jure Leskovec. Community detection in networks with node attributes. In *Data Mining (ICDM), 2013 IEEE 13th international conference on*, pages 1151–1156. IEEE, 2013.

Tianbao Yang, Rong Jin, Yun Chi, and Shenghuo Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 927–936. ACM, 2009.