Research Article

Differentially Private Attributed Network Releasing Based on Early Fusion

Yuye Wang , Jing Yang , and Jianpei Zhan

College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang, China

Correspondence should be addressed to Jing Yang; yangjing@hrbeu.edu.cn

Received 4 April 2021; Accepted 14 July 2021; Published 27 July 2021

Academic Editor: Bruce M. Kapron

Copyright © 2021 Yuye Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Vertex attributes exert huge impacts on the analysis of social networks. Since the attributes are often sensitive, it is necessary to seek effective ways to protect the privacy of graphs with correlated attributes. Prior work has focused mainly on the graph topological structure and the attributes, respectively, and combining them together by defining the relevancy between them. However, these methods need to add noise to them, respectively, and they produce a large number of required noise and reduce the data utility. In this paper, we introduce an approach to release graphs with correlated attributes under differential privacy based on early fusion. We combine the graph topological structure and the attributes together with a private probability model and generate a synthetic network satisfying differential privacy. We conduct extensive experiments to demonstrate that our approach could meet the request of attributed networks and achieve high data utility.

1. Introduction

Social network is an important topic in many subjects, such as sociology, economics, and informatics. Many applications need to achieve useful information through analyzing social networks. Social networks contain sensitive personal information, including the individual information and the relationships among them. However, vertices usually contain sensitive attribute information and we need to protect the privacy of not only the graph topological structure but also the vertex attributes. We call the network with vertex attributes as attributed network. In this paper, we study the privacy protection method of releasing the attributed networks. Network data can be released for public with sanitization by a substituted synthetic network generated from a private network generation model.

As a common privacy protection method, anonymization technique (e.g., k -anonymity [1] and l -diversity [2]) could easily be susceptible to new privacy attacks. Recently, differential privacy has been proposed to address such privacy problem. Unlike the anonymization technique, differential privacy provides strong theoretical guarantees by adding calibrated noise against adversaries with prior knowledge.

The standard technique to ensure differential privacy of social network is to “sanitize” the topological structure and neglect the vertex attributes. These vertex attributes affect not only the vertex information but also the topological structure. For example, vertex attributes are closely related to the connection between vertices. When the topological graph has high assortativity coefficient, vertices with similar attributes connect with each other with a higher probability. Adversaries with vertex attributes could infer the topological structure effectively.

A straightforward approach to release the attributed network under differential privacy is to sanitize the vertex attributes and topological structure, respectively, and then fuse them together to get the differentially private version of the network. We call this kind of method as *late fusion*. On the contrary, when we deal with the vertex attributes and topological structure simultaneously, we call it as *early fusion*. In this paper, we study the attributed network releasing method under differential privacy through early fusion.

We propose a probability model of attributed network under differential privacy, called DP-ANP (i.e., differentially private attributed network probability model). DP-ANP contains the vertex attribute information and the topological structure. The DP-ANP records connection probabilities

between any pair of vertices and attribute value probabilities of each vertex. It allows to draw a sample model from the model's space through joint probability distribution, and it is equivalent to maximizing the probability of the model. In particular, to satisfy the differential privacy, we add noise to the parameters of the probability model and generate the private synthetic network using DP-ANP.

The core of DP-ANP is to add noise to the model parameters of each vertex. However, due to the large scale of vertices, DP-ANP may lead to a large amount of noise. Therefore, we propose an improved DP-ANP, called DP-ANPHP (i.e., differentially private attributed network probability model based on hyperparameter). In DP-ANPHP, we construct a model of network based on hyperparameters and add noise to them instead of the model parameters in DP-ANP. The scale of hyperparameters in DP-ANPHP is much less than the parameters in DP-ANP. It makes DP-ANPHP produce less noise and achieve high data utility. In summary, we present several contributions:

- (1) We introduce the probability model of attributed network under differential privacy based on early fusion, called DP-ANP. This model sanitizes the model parameters of the overall information of the network instead of vertex attributes and topological structure separately. Compared with the late fusion method, DP-ANP reduces the complexity of computing.
- (2) We develop an improved DP-ANP model based on hyperparameters, called DP-ANPHP. DP-ANPHP reduces the scale of noise.
- (3) Through privacy analysis, we prove that DP-ANP and DP-ANPHP satisfy differential privacy. We conduct extensive experimental study over synthetic datasets and real datasets. The results demonstrate that DP-ANP outperforms the late fusion method and DP-ANPHP outperforms DP-ANP.

The rest of the paper is organized as follows. Section 2 provides a literature review of differential privacy of social network. Section 3 presents necessary concept on differential privacy and attributed network. Section 4 describes early fusion and late fusion. Section 5 describes the DP-ANP model. Section 6 describes the DP-ANPHP model. Section 7 reports the experimental results. Section 8 concludes the paper.

2. Related Work

2.1. Differentially Private Network Analysis. Dwork et al. [3] firstly proposed a differentially private method to answer the query functions by disturbing the outcome directly. Hay et al. [4] proposed a differential privacy in a post-processing phase to compute the output which retained the accuracy and privacy at the same time. They used this method to estimate the degree distribution. Nissim et al. [5] introduced the concept of smooth sensitivity and protected individuals' privacy by adding a small amount of noise to the released statistics, such as triangle count. Karwa et al. [6] expanded

this concept and calculate k -star count of a network. Zhang et al. [7] analyzed the released statistics through a ladder function and reduced the sensitivity effectively. Cheng et al. [8] presented a two-phase differentially private frequent subgraph mining algorithm, named DFG. In DFG, frequent subgraphs are privately identified in the first phase, and the noisy support of each identified frequent subgraph is calculated in the second phase. Ding et al. [9] published the private triangle counts with the triangle count distribution and the cumulative distribution. Sun et al. [10] studied fundamental problems related to extended local view. They formulated a decentralized private scheme named DDP, which requires that each participant consider both her own privacy and her neighbors' privacy involved in her ELV.

2.2. Differentially Private Network Publishing. Sala et al. [11] proposed a private graph model which extracts a graph's structure into dK-graph and generated a synthetic graph. Mir and Wright [12] used maximum likelihood estimation to privately estimate the parameters based on the stochastic Kronecker graph model. Xiao et al. [13] proposed a private network publishing method which computes an estimator of the graph in the hierarchical random graph (HRG) model. They sampled possible HRG structures via Markov chain Monte Carlo (MCMC) through exponential mechanism. Qin et al. [14] investigated techniques to ensure local differential privacy of individuals while collecting structural information and generating representative synthetic social graphs. They proposed LDPGen which incrementally clusters users based on their connections to different partitions of the whole population and adapted existing social graph generation models to construct a synthetic social graph. Chen et al. [15] presented a method for publishing private synthetic graphs, which preserves the community structure of the original graph without sacrificing the ability to capture global structural properties. Wang et al. [16] presented a differential privacy method for weighted network through structuring a private probability model.

2.3. Differential Privacy for Attributed Network. Qian et al. [17] proposed to deanonymize social graphs and infer private attributes leveraging knowledge graphs, which carry both graph structural information and semantic information. Ji et al. [18] conducted the first attribute-based anonymity analysis for attributed network under both preliminary and general models and proposed a new deanonymization framework, which takes into account both the graph structure and the attribute information to the best of our knowledge. Jorgensen et al. [19] proposed a method for publishing private attributed network which adapted existing graph models and introduced a new one and then showed how to augment them with differential privacy. Yin et al. [20] defined a new type of attack called attribute couplet attack and proposed a new anonymity concept called k -couplet anonymity to achieve privacy preservation under attribute couplet attacks. Kiranmayi et al. [21] designed an algorithm named couplet anonymization by using the node addition approach to reduce the misleading fake relations.

Meanwhile, some research studies focus on the privacy of high-dimensional data. These methods are also closely related to the differential privacy for attributed network. Chen et al. [22] proposed a novel solution to preserve the joint distribution of a high-dimensional dataset. They developed a robust sampling-based framework to systematically explore the dependencies among all attributes and identified a set of marginal tables from the dependency graph to approximate the joint distribution based on the solid inference foundation of the junction tree algorithm. Zhang et al. [23] presented a differentially private method for releasing high-dimensional data named PrivBayes. PrivBayes injected noise into the low-dimensional marginals instead of the high-dimensional dataset to circumvent the curse of dimensionality. They also introduced a novel approach that uses a surrogate function for mutual information to build the model more accurately. Ren et al. [24] developed a local differentially private high-dimensional data publication algorithm, LoPub, by taking advantage of our distribution estimation techniques. Correlations among multiple attributes were identified to reduce the dimensionality of crowdsourced data.

In conclusion, only a few number of research studies focus on differential privacy for attributed network and are generally based on late fusion. In this work, we develop a differentially private attributed network publishing method based on early fusion.

3. Background

3.1. Differential Privacy. Given a graph G , differential privacy ensures the outputs to be approximately same even if any edge is arbitrarily added or deleted in the graph. Thus, the presence or absence of any edge has a negligible effect on the outputs. We define two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ to be neighbors if they satisfy $V_1 = V_2$, $E_1 \subset E_2$ and $|E_2| = |E_1| + 1$. ϵ -Differential privacy is defined as follows.

Definition 1 (ϵ -differential privacy) (see [3]). A randomized algorithm A is ϵ -differential privacy if for any two neighboring graphs G_1 and G_2 , and for any output $O \in \text{Range}(A)$,

$$\Pr[A(G_1) \in O] \leq e^\epsilon \Pr[A(G_2) \in O]. \quad (1)$$

Differential privacy is based on the concept of *global sensitivity* of a function f . It is used to measure the maximum change in the outputs of f when any edge in the graph is changed. The global sensitivity of f is defined as $\Delta f = \max_{G_1, G_2} \|f(G_1) - f(G_2)\|_1$.

Differential privacy can be achieved by Laplace mechanism and exponential mechanism. The Laplace mechanism is mainly used for functions whose outputs are real values. Differential privacy can be achieved by adding properly noise drawn randomly from Laplace distribution to the true answer.

Theorem 1 (laplace mechanism) (see [3]). *For any function $f: G \rightarrow \mathbb{R}^d$ with sensitivity Δf , the algorithm*

$$A(G) = f(G) + \langle \text{Lap}_1\left(\frac{\Delta f}{\epsilon}\right), \dots, \text{Lap}_d\left(\frac{\Delta f}{\epsilon}\right) \rangle, \quad (2)$$

satisfies ϵ -differential privacy, where $\text{Lap}_i(\Delta f/\epsilon)$ are i.i.d Laplace variables with scale parameter $\Delta f/\epsilon$.

The exponential mechanism is mainly used for functions whose outputs are not real numbers. The main idea is to sample the output data O from the output space O according to the utility function u . The global sensitivity of u is $\Delta u = \max_{O_1, O_2} |u(G_1, O_1) - u(G_2, O_2)|$.

Theorem 2 (exponential mechanism) (see [25]). *Given a graph G and a utility function $u: (G \times O) \rightarrow \mathbb{R}$, the arithmetic A whose output is with probability proportional to $\exp(\epsilon \cdot u(G, O)/2\Delta u)$ satisfies ϵ -differential privacy.*

Theorem 3 (sequential composition 1) (see [26]). *If each arithmetic A_i provides ϵ_i -differential privacy, a sequence of $(A_1(D), A_2(D), \dots, A_n(D))$ over (A) the same database D provides $\sum_{i=1}^n \epsilon_i$ -differential privacy.*

Theorem 4 (sequential composition 2) (see [27]). *Any subset D_{iter} sampled from D satisfying each data point is included independently with probability p . If algorithm $A(D_{\text{iter}})$ satisfies ϵ_{iter} -differential privacy, $A(D)$ satisfies $\log(1 + p((e^{\epsilon_{\text{iter}}} - 1)))$ -differential privacy.*

3.2. Attributed Network. Attributed network G is denoted by the triple (V, E, Λ) . $V = \{v_1, v_2, \dots, v_N\}$ is a set of vertices, $E = \{e_{ij} = (v_i, v_j) | 1 \leq i, j \leq N, i \neq j\}$ is a set of edges, and $\Lambda = \{a^1, a^2, \dots, a^S\}$ is the set of attribute S . Given the integer K , N vertices are divided into K disjoint subsets. Let e_{ij} be an undirected edge and G be an undirected attributed network.

4. Early Fusion and Late Fusion

We classify the methods of publishing differentially private attribute network into two categories: early fusion and late fusion.

The main idea of early fusion is constructing a private network model which combines the vertex attributes and topological structure into a whole structure and allows two parts to interact with each other. Early fusion makes the network model private and uses it to release the private vertex attributes and topological structure. Figure 1 shows the structure of early fusion.

Late fusion needs to consider that vertex attributes affect the topological structure. Adversaries might pose threats to the privacy through the correlation between them without considering that vertex attributes affect the topological structure. As a result, late fusion needs to design the parameters which represent the correlation between vertex attributes and topological structure and sanitize them. However, it results in additional privacy cost and increases the complexity of the algorithm. Figure 2 shows the structure of late fusion.

Compared with late fusion, early fusion constructs the network model and combines the vertex attributes and

FIGURE 1: The structure of early fusion.

FIGURE 2: The structure of late fusion.

topological structure into a whole structure which contains the correlation. There is no need to design the correlation parameters. In this paper, we study the attributed network releasing method under differential privacy through early fusion.

5. Differentially Private Attributed Network Model

In this section, we introduce a kind of attributed network probability (ANP) model and enable generation of synthetic attributed graph with differential privacy.

5.1. Privacy Analysis of Straightforward Approach. In practice, we know that vertices with similar attributes connect with each other with a higher probability. For example, in social network, people with similar interest in a certain field may follow a well-known person in that field in common. They are all connected with this vertex in social network. To simplify the complexity, we divide these vertices into the

same group. Vertices in the same group are stochastically equivalent. They have similar attributes and play equivalent roles in generating the network's structure. In particular, they have the attributes which obey the same distribution and connect with other vertices with the same distribution. We assume that this kind of partition exists but is unknowable. Thus, it can be treated as hidden variable and cannot be observed directly.

ANP is denoted by a triple (X, Y, Z) . Adjacency matrix $X = [X_{ij}]$ is an $N \times N$ symmetric random matrix, where X_{ij} is a binary random variable which takes value from $\{0, 1\}$. When there is an edge between vertices v_i and v_j , $X_{ij} = 1$; otherwise, $X_{ij} = 0$. Attribute matrix $Y = [Y_i^s]$ is an $N \times S$ matrix, where Y_i^s is a random variable which denotes the value of attribute s associated with vertex v_i . Group matrix $Z = [Z_i]$ is an $N \times I$ matrix, where Z_i is a random variable denoting the label of the group which v_i belongs to. ANP outputs a sample from all possible attributed graphs. ANP uses the parameters α , θ , and ϕ ; a compact graphical representation is given in Figure 3.

FIGURE 3: A graphical representation of ANP.

The label of the group Z_i is sampled from a multinomial distribution with its parameter α representing the probability of vertex belonging to the group. It is defined as

$$\Pr(Z_i = k|\alpha) = \alpha_k, \quad k = 1, 2, \dots, K, \quad (3)$$

where α is a K -vector parameter and $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_K)$. α_k denotes the proportion of vertex belonging to the group k and satisfies $\alpha_k \in [0, 1]$ and $\sum_{k=1}^K \alpha_k = 1$.

Suppose vertices in the same group have similar attributes and their attributes are samples from the same distribution. Given the label of the group Z_i , the attribute Y_i^t is sampled from a multinomial distribution with its parameter $\theta_{Z_i}^t$ representing the probability of attribute values. It is defined as

$$\Pr(Y_i^s = m|\theta_{Z_i}^s) = \theta_{Z_i,m}^s, \quad m = 1, 2, \dots, M, \quad (4)$$

where $\theta_{Z_i}^s$ is a M -vector parameter and $\theta_{Z_i}^s = (\theta_{Z_i,1}^s, \theta_{Z_i,2}^s, \dots, \theta_{Z_i,M}^s)$. If vertex i belongs to Z_i , $\theta_{Z_i}^s$ denotes the proportion of vertex in Z_i values as m and satisfies $\theta_{Z_i,m}^s \in [0, 1]$ and $\sum_{m=1}^M \theta_{Z_i,m}^s = 1$.

Suppose vertices from Z_i and vertices from Z_j are connected with each other from the same distribution. Given the label of the group Z_i , X_{ij} is sampled from a Bernoulli distribution with its parameter $\phi_{Z_i Z_j}$ representing the probability of whether there is an edge between vertices Z_i and Z_j . It is defined as

$$\Pr(X_{ij} = 1|\phi_{Z_i Z_j}) = \phi_{Z_i Z_j}^{X_{ij}} \cdot (1 - \phi_{Z_i Z_j})^{1 - X_{ij}}, \quad (5)$$

where $\phi_{Z_i Z_j}$ satisfies $\phi_{Z_i Z_j} \in [0, 1]$ and $\phi_{Z_i Z_j} = \phi_{Z_j Z_i}$.

We use maximum likelihood estimation (MLE) to specify the values of α , θ , and ϕ . The group label Z is treated as hidden variable and cannot be observed directly. Adjacency matrix X and attributes Y can be observed directly and could be used for the calculation. We take the expectation maximization (EM) algorithm to address this issue:

E step: estimate the distribution of hidden variable $\Pr(Z|X, Y, \{\alpha, \theta, \phi\}^t)$.

M step: get the parameters through finding the maximize expectations, and the parameters are computed as follows:

$$\{\alpha, \theta, \phi\}^{t+1} = \arg \max_{(\alpha, \theta, \phi)} \sum_Z \Pr(Z|X, Y, \{\alpha, \theta, \phi\}^t) \cdot \log \Pr(X, Y, Z|\{\alpha, \theta, \phi\}^t). \quad (6)$$

5.2. Differentially Private ANP. Our goal is to publish a differentially private attributed network $\tilde{G}$ which matches the topological structure and attribute distribution of the original network as much as possible. We identify the private ANP model and generate $\tilde{G}$ from it.

To identify the private ANP model, we need to add noise to the three kinds of parameters and get the noisy parameters. Then, we use the noisy parameters to generate the sanitized network $\tilde{G}(\tilde{X}, \tilde{Y}, \tilde{Z})$. Our differentially private attributed network publishing algorithm is shown in Algorithm 1.

To satisfy differential privacy, we need to calculate the noisy parameters during each iteration process of parameter estimation. We add noise to all the parameters and the scale of noise depends on the sensitivity of the parameters. In each iteration process, we run the EM algorithm and obtain the parameters $\{\alpha, \theta, \phi\}$ (lines 4–5). We firstly add noise to the group parameter α (line 6). α_k denotes the proportion of the vertices belonging to group k , and the sensitivity of α is calculated as the maximum change in α_k if any one vertex changes its group label. Suppose the number of vertices in group k is l_k , we have $\alpha_k = l_k/N$. If one vertex from group k changes its group label to other groups or one vertex from other group changes its group label to k , α_k changes to $(l_k -$

$1)/N$ or $(l_k + 1)/N$. The global sensitivity of α is $1/N$. Then, we sample $\tilde{\alpha}$ through the multinomial distribution with the parameter $\tilde{\alpha}$ (line 6) and add noise to the attribute parameter θ (line 8). The parameter θ of each group is different to each other. $\theta_{Z_i,m}$ denotes the proportion of vertices in group Z_i that take the attribute value as m , and the sensitivity of θ is calculated as the maximum change in $\theta_{Z_i,m}$ if any one vertex in group Z_i changes its attribute. Suppose the number of vertices with the attribute equal to m in group Z_i is q and the number of vertices in group Z_i is l_{Z_i} (line 7), we have $\theta_{Z_i,m} = q/l_{Z_i}$. If one vertex from group Z_i changes its attribute m to other value or one vertex changes its attribute value to m , $\theta_{Z_i,m}$ changes to $(q - 1)/l_{Z_i}$ or $(q + 1)/l_{Z_i}$. The global sensitivity of θ is $1/l_{Z_i}$. At last, we add noise to the parameter ϕ (line 9). The sensitivity of ϕ is calculated as the maximum change in $\phi_{Z_i Z_j}$ if any one vertex disappears or appears. Suppose the number of vertices in group Z_i and Z_j is l_{Z_i} and l_{Z_j} , respectively, and there are r edges between group Z_i and Z_j , we have $\phi_{Z_i Z_j} = 2r/l_{Z_i} \cdot l_{Z_j}$. If one vertex from group Z_i connects with s vertices in group Z_j , after deleting this vertex, we have $\phi_{Z_i Z_j} = 2(r - s)/l_{Z_i} \cdot l_{Z_j}$. The maximum change of $\phi_{Z_i Z_j}$ is $2s/l_{Z_i} \cdot l_{Z_j}$. When this vertex connects all the vertices in group Z_j , s takes the maximum value, and $s = l_{Z_j}$. Thus, the global sensitivity of ϕ is $2/l_{Z_i}$.

Input: node dataset V , attribute dataset Λ , group number K , privacy parameter $\varepsilon_1, \varepsilon_2, \varepsilon_3$, maximum iterations J .
Output: sanitized attribute network $\tilde{G}(\tilde{X}, \tilde{Y}, \tilde{Z})$.

- (1) Initialize the attribute network (X, Y, Z) from V, Λ , and K .
- (2) Initialize the parameters $\{\alpha, \theta, \phi\}^0$
- (3) **for** $t = 1, \dots, J$ **do**
- (4) **E step:** given $\{\alpha, \theta, \phi\}^{t-1}$, compute $\log \Pr(X, Y, Z | \{\alpha, \theta, \phi\}^{t-1})_{\Pr(Z|X, Y, \{\alpha, \theta, \phi\}^{t-1})}$.
- (5) **M step:** compute $\{\alpha, \theta, \phi\}^t = \operatorname{argmax}_{(\alpha, \theta, \phi)} \sum_Z \Pr(Z | X, Y, \{\alpha, \theta, \phi\}^{t-1}) \cdot \log \Pr(X, Y, Z | \{\alpha, \theta, \phi\}^{t-1})$.
- (6) Compute $\tilde{\alpha}^t = \alpha^t + \operatorname{Lap}(1/N\varepsilon_1)$ and choose $\tilde{Z}_i^t \sim \operatorname{Multinomial}(\tilde{\alpha}^t)$.
- (7) Given $\tilde{Z}_i^t$, compute the number of nodes from $\tilde{Z}_i^t$, called $l_{Z_i}^t$.
- (8) Compute $\tilde{\theta}_{Z_i}^t = \theta_{Z_i}^t + \operatorname{Lap}(1/\varepsilon_2 l_{Z_i}^t)$.
- (9) Compute $\tilde{\phi}_{Z_i, Z_j}^t = \phi_{Z_i, Z_j}^t + \operatorname{Lap}(2/\varepsilon_3 l_{Z_i}^t)$.
- (10) **end for**
- (11) Choose $\tilde{Y}_i \sim \operatorname{Multinomial}(\tilde{\theta}_{Z_i})$.
- (12) **for** each vertex $v_i \in V$ and $i > j$ **do**
- (13) choose $\tilde{X}_{ij} \sim \operatorname{Bernoulli}(\tilde{\phi}_{Z_i, Z_j})$.
- (14) **end for**
- (15) **return** $\tilde{G}(\tilde{X}, \tilde{Y}, \tilde{Z})$.

ALGORITHM 1: Differentially private attributed network publishing by ANP (DP-ANP).

After getting the noisy parameters $\{\tilde{\alpha}, \tilde{\theta}, \tilde{\phi}\}$, we sample vertex attribute from the multinomial distribution with the parameter $\tilde{\theta}$ (line 11) and sample edges from the Bernoulli distribution with the parameter $\tilde{\phi}$ (lines 12–14).

5.3. Privacy Analysis of DP-ANP. We prove that DP-ANP satisfies ε -differential privacy based on the sequential composition property.

Theorem 5. *DP-ANP satisfies ε -differential privacy.*

Proof. Suppose that DP-ANP makes the parameters converge after iterating J times. From Algorithm 1, we know that during each iteration process, $\tilde{\alpha}$ satisfies ε_1 -differential privacy, $\tilde{\theta}$ satisfies ε_2 -differential privacy, and $\tilde{\phi}$ satisfies ε_3 -differential privacy. Based on the sequential composition property, J times iteration process satisfies $J \cdot (\varepsilon_1 + \varepsilon_2 + \varepsilon_3)$ -differential privacy. Furthermore, generating synthetic network does not consume any privacy budget. In conclusion, DP-ANP satisfies ε -differential privacy and $\varepsilon = J \cdot (\varepsilon_1 + \varepsilon_2 + \varepsilon_3)$. $\square$

6. Differentially Private Attributed Network Model Based on Hyperparameter

6.1. Differentially Private Attributed Network Model Based on Hyperparameter (DP-ANPHP). As discussed in the previous section, we need to add all the parameters α, θ, ϕ to satisfy

differential privacy in DP-ANP. To reduce the amount of noise and relieve the effect of the noise, we propose a differentially private attributed network publishing method based on hyperparameter, called DP-ANPHP. In DP-ANPHP, we learn the parameters by Bayesian theory and propose an efficient differential privacy method by sanitizing a part of hyperparameters. Thus, the scale of parameters we need to sanitize decreases, and we could generate synthetic attributed network effectively. At last, we prove that DP-ANPHP satisfies differential privacy.

However, some difficulties lie in the calculation of the posterior distribution of hidden variable when we use probabilistic inference model in Bayesian theory. Therefore, we use variational Bayesian algorithm to address this problem. We treat parameters α, θ , and ϕ as random variables, and let them have prior distribution. We place a Dirichlet distribution over α as the prior distribution with hyper-parameter ξ , and we also place a Dirichlet distribution θ as the prior distribution with hyper-parameter γ . We place Beta distribution over ϕ as the prior distribution with hyper-parameter (μ, ν) . As Dirichlet distribution and Beta distribution are the conjugate priors for multinomial distribution and Bernoulli distribution, respectively, the posterior distribution of α and θ is still Dirichlet distribution, and the posterior distribution of θ is still Beta distribution. This will result in mathematical convenience.

Let $q(z, \alpha, \theta, \phi)$ be the variational distribution to approximate the distribution $\Pr(Z, \alpha, \theta, \phi | X, Y)$, and the constant log-likelihood $\log \Pr(X, Y)$ can be represented as

$$\begin{aligned} \log \Pr(X, Y) &= \iiint \sum_z q(z, \alpha, \theta, \phi) \log \Pr(X, Y) \, d\alpha d\theta d\phi \\ &= \iiint \sum_z q(z, \alpha, \theta, \phi) \log \frac{\Pr(X, Y, Z, \alpha, \theta, \phi)}{q(z, \alpha, \theta, \phi)} \, d\alpha d\theta d\phi \end{aligned}$$

$$\begin{aligned}
& - \iiint \sum_z q(z, \alpha, \theta, \phi) \log \frac{\Pr(Z, \alpha, \theta, \phi | X, Y)}{q(z, \alpha, \theta, \phi)} d\alpha d\theta d\phi \\
& = \mathcal{G}(q) + \text{KL}(\Pr(Z, \alpha, \theta, \phi | X, Y) \| q(z, \alpha, \theta, \phi)),
\end{aligned} \tag{7}$$

where we adopt the Kullback–Leibler (KL) divergence to measure the distance between $\Pr(Z, \alpha, \theta, \phi | X, Y)$ and $q(z, \alpha, \theta, \phi)$:

$$\text{KL}(\Pr(Z, \alpha, \theta, \phi | X, Y) \| q(z, \alpha, \theta, \phi)) = - \iiint \sum_z q(z, \alpha, \theta, \phi) \log \frac{\Pr(Z, \alpha, \theta, \phi | X, Y)}{q(z, \alpha, \theta, \phi)} d\alpha d\theta d\phi. \tag{8}$$

Let $\mathcal{G}(q)$ to be the functional lower bound on the $\log \Pr(X, Y)$:

$$\mathcal{G}(q) = \iiint \sum_z q(z, \alpha, \theta, \phi) \log \frac{\Pr(X, Y, Z, \alpha, \theta, \phi)}{q(z, \alpha, \theta, \phi)} d\alpha d\theta d\phi. \tag{9}$$

Thus, minimizing the $\text{KL}(\Pr(Z, \alpha, \theta, \phi | X, Y) \| q(z, \alpha, \theta, \phi))$ is equivalent to maximizing $\mathcal{G}(q)$. When we maximize $\mathcal{G}(q)$, $q(z, \alpha, \theta, \phi)$ gets closer to $\Pr(Z, \alpha, \theta, \phi | X, Y)$.

Furthermore, the variational distribution $q(z, \alpha, \theta, \phi)$ can be factorized as

$$q(z, \alpha, \theta, \phi) = q(\alpha)q(\theta)q(\phi) \prod_i q(Z_i), \tag{10}$$

where $i = 1, 2, \dots, N$. We further convert the variational distribution to the hyperparameter form. Based on the conjugate distribution, $q(\alpha|\xi)$ and $q(\theta|\gamma)$ are defined as Dirichlet distribution with hyperparameters ξ and γ , respectively, and $q(\phi|\mu, \nu)$ is defined as beta distribution with

hyperparameter (μ, ν) . And we also introduce a new variational parameter β_i to replace α as the parameter of group label Z_i . The reason is that according to equation (10), we require Z_i to no longer depend on α . So, we treat β_i as the hyperparameter of Z_i . q has the following form:

$$q(z, \alpha, \theta, \phi | \xi, \gamma, \mu, \nu, \beta_i) = q(\alpha|\xi)q(\theta|\gamma)q(\phi|\mu, \nu) \prod_i q(Z_i|\beta_i). \tag{11}$$

To maximize $\mathcal{G}(q)$, we take the derivatives of $\mathcal{G}(q)$ with respect to ξ, γ, μ, ν , and β and set them to zeros:

$$\xi_k^{t+1} = \xi_k^t + \sum_{i=1}^N \beta_{ik}^{t+1}, \tag{12}$$

$$\gamma_{km}^{t+1} = \gamma_{km}^t + \sum_{i=1}^N \delta(Y_i, a_m) \beta_{ik}^{t+1}, \tag{13}$$

$$\mu_{kl}^{t+1} = \mu_{kl}^t + \sum_{\substack{i,j=1 \\ i \neq j}}^N X_{ij} \beta_{ik}^{t+1} \beta_{jk}^{t+1}, \tag{14}$$

$$\nu_{kl}^{t+1} = \nu_{kl}^t + \sum_{\substack{i,j=1 \\ i \neq j}}^N (1 - X_{ij}) \beta_{ik}^{t+1} \beta_{jk}^{t+1}, \tag{15}$$

$$\beta_{ik}^{t+1} \propto \exp \left\{ f(\xi_k^{t+1}) + \sum_{m=1}^M \delta(Y_i, a_m) g(\gamma_{km}^{t+1}) + \sum_{j=1}^N \sum_{l=1, l \neq i}^K \beta_{jl}^t X_{ij} h(\mu_{kl}^{t+1}, \nu_{kl}^{t+1}) \right\}, \quad (16)$$

where $i = 1, 2, \dots, N$; $k = 1, 2, \dots, K$; $l = k + 1, k + 2, \dots, K$; $m = 1, 2, \dots, M$. $\delta(Y_i, a_m)$ is Kronecker delta function. $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are the linear functions with parameters ξ_k^{t+1} , γ_{km}^{t+1} , and $(\mu_{kl}^{t+1}, \nu_{kl}^{t+1})$, respectively.

To satisfy differential privacy, we need to add noise to hyperparameters during each iteration process. In practice, we found that the functions of ξ , γ , μ , and ν are with respect to the parameter β . The coefficients of β are $\delta(Y_i, a_m)$, X_{ij} , and $(1 - X_{ij})$, respectively. They are set to 1 or 0, and we could treat them as the range of the sum of β . Hence, we only need to add noise to β during the iteration process to satisfy differential privacy. In this way, we could reduce the scale of parameters which we need to add noise to and reduce the amount of noise required by differential privacy. Similar to the parameter α in DP-ANP, the global sensitivity of β is $1/N$. The differentially private attributed network publishing algorithm based on hyperparameter is shown in Algorithm 2.

6.2. Privacy Analysis of DP-ANPHP. Based on the sequential composition property, we prove that DP-ANPHP satisfies ϵ -differential privacy.

Theorem 6. *DP-ANPHP satisfies ϵ -differential privacy.*

Proof. Suppose that DP-ANPHP makes the parameters converge after iterating J times. From Algorithm 2, we know that during each iteration process, $\tilde{\beta}$ satisfies ϵ_1 -differential privacy. Based on the sequential composition property, J times iteration process satisfies $J \cdot \epsilon_1$ -differential privacy. Furthermore, generating synthetic network does not consume any privacy budget. In conclusion, DP-ANPHP satisfies ϵ -differential privacy and $\epsilon = J \cdot \epsilon_1$. $\square$

7. Experimental Results

Existing work on differentially private attributed network publishing algorithm has mainly focused on the late fusion method. For comparing the results effectively, Algorithm 3 shows a kind of late fusion method to compare DP-ANP and DP-ANPHP. Firstly, we need to add noise to the connection probability directly (line 9) and perturb the attributes using exponential mechanism (line 12). We emphasize that these two steps are independent.

We evaluate the performance of DP-ANP and DP-ANPHP with LN. As the Laplace mechanism differential privacy produces random noise, we measure the accuracy of the result using the median relative error where we run the Laplace mechanism for 10 times.

7.1. Datasets. We evaluate the utility over one synthetic dataset and two real-life datasets. The synthetic data consist of 100 vertices, and there are a number of edges randomly between them. It have two attributes. One includes two attribute values, and the other one includes three attribute values. Real-life datasets are Cora and Flickr. As these two datasets contain a large number of attributes, we sample 5 discrete attributes as our experimental data. The statistics of datasets are given in Table 1. The fill of a network is the proportion of edges to the number of all possible edges, and it denotes the probability that an edge is present between two randomly vertices.

7.2. Evaluation under Different ϵ . We compare the utility of LN, DP-ANP, and DP-ANPHP through the measure average clustering coefficient, number of triangles, normalized degree distribution, and modularity. We fix K at 4 and allocate the privacy budget ϵ as follows: 0.25, 0.5, 0.75, and 1. We represent the relative error of these measurements as

$$\text{rel.err} = \frac{|A - \tilde{A}|}{A}, \quad (17)$$

where $\tilde{A}$ is the differentially private output.

In Figures 4 and 5, we compare the correlated error of average clustering coefficient and number of triangles. We can see that DP-ANP and DP-ANPHP outperform LN, and DP-ANPHP outperforms DP-ANP. This means that DP-ANPHP performs better in terms of representing the density level. Furthermore, with the increase of ϵ , the correlated error of DP-ANPHP had greater decreases, and the decrease of LN and DP-ANP is smaller. The reason is that in DP-ANPHP, we only need to add noise to the hyperparameter β . When the private strength is small, the noise affects the output more slightly. We need to add noise to all the parameters of the model in LN and to add noise to a part of the parameters in DP-ANP. The noise affects the output more greatly. In Figure 6, we compare the correlated error of normalized degree distribution. We can see that DP-ANPHP outperforms DP-ANP and DP-ANP outperforms LN.

We use the modularity as the measure of cluster analysis to compare DP-ANPHP with DP-ANP. Modularity is a measurement for evaluating the result of community division. It reflects the concentration of nodes in the same group. The higher the modularity, the tighter the nodes in the same group and the looser the nodes in different groups. Modularity has a range of $[-1, 1]$. In Figure 7, we can see that the modularity of DP-ANPHP is higher than that of DP-ANP. This is mainly

Input: node dataset V , attribute dataset Λ , group number K , privacy parameter ϵ_1 , maximum iterations J .

Output: sanitized attribute network $(\tilde{X}, \tilde{Y}, \tilde{Z})$.

- (1) Initialize the attribute network (X, Y, Z) from V, Λ , and K .
- (2) Initialize β^0 .
- (3) **for** $t = 1, \dots, J$ **do**
- (4) Given $\tilde{\beta}^{t-1}$, compute $\{\xi, \gamma, \mu, \nu\}^t$ from equations (12)–(15).
- (5) Given $\{\xi, \gamma, \mu, \nu\}^t$, compute β^t from equation (16).
- (6) Compute $\tilde{\beta}^t = \beta^t + \text{Lap}(1/\epsilon_1 N)$.
- (7) **end for**
- (8) Choose $\alpha \sim \text{Dirichlet}(\xi)$, $\theta \sim \text{Dirichlet}(\gamma)$, $\phi \sim \text{Beta}(\mu, \nu)$.
- (9) Choose $\tilde{Z} \sim \text{Multinomial}(\alpha)$, $\tilde{Y} \sim \text{Multinomial}(\theta)$, $\tilde{X} \sim \text{Bernoulli}(\phi)$.
- (10) **return** $(\tilde{X}, \tilde{Y}, \tilde{Z})$.

ALGORITHM 2: Differentially private attributed network publishing algorithm based on hyperparameter (DP-ANPHP).

Input: node dataset V , attribute dataset Λ , group number K , privacy parameter $\epsilon_1, \epsilon_2, \epsilon_3$, maximum iterations J .

Output: sanitized attribute network $\tilde{G}(\tilde{X}, t\tilde{Y}n, q\tilde{Z})$.

- (1) Initialize the attribute network (X, Y, Z) from V, Λ and K .
- (2) Initialize the parameters $\{\alpha, \phi\}^0$.
- (3) **for** $t = 1, \dots, J$ **do**
- (4) **E step:** given $\{\alpha, \phi\}^{t-1}$, compute $\log \Pr(X, Z | \{\alpha, \phi\}^{t-1})_{\Pr(Z|X, \{\alpha, \phi\}^{t-1})}$.
- (5) **M step:** compute $\{\alpha, \phi\}^t =$
- (6) $\text{argmax}_{(\alpha, \phi)} \sum_Z \Pr(Z | X, \{\alpha, \phi\}^{t-1}) \cdot \log \Pr(X, Z | \{\alpha, \phi\}^{t-1})$.
- (7) Compute $\tilde{\alpha}^t = \alpha^t + \text{Lap}(1/N\epsilon_1)$ and choose $\tilde{Z}_i^t \sim \text{Multinomial}(\tilde{\alpha}^t)$.
- (8) Given $\tilde{Z}_i^t$, compute the number of vertices from $\tilde{Z}_i^t$, called $l_{Z_i}^t$.
- (9) Compute $\tilde{\phi}_{Z_i Z_j}^t = \phi_{Z_i Z_j}^t + \text{Lap}(2/\epsilon_3 l_{Z_i}^t)$.
- (10) **end for**
- (11) Compute $u = n_m/N$, where n_m denotes the number of the vertices which take the attribute value as m .
- (12) Compute attribute parameter $\tilde{\theta} = \exp(\epsilon_2 u / 2\Delta u)$, where Δu denotes the sensitivity of u .
- (13) Compute correlation parameter Θ using $\tilde{\phi}$ and $\tilde{\theta}$.
- (14) Sample attribute $\tilde{Y}_i$ with probability $\tilde{\theta}$.
- (15) Sample edge $\tilde{X}_{ij}$.

ALGORITHM 3: Differentially private attributed network publishing by late fusion (LN).

TABLE 1: Synthetic data and Cora and Flickr statistics.

Dataset	#Nodes	#Edges	#Attributes	Fill
Synthetic data	100	2705	2	0.2705
Cora	2708	10556	1433	0.0014
Flickr	7575	479476	12047	0.00835

FIGURE 4: Relative error of average clustering coefficient under different ϵ . (a) Synthetic data. (b) Cora. (c) Flickr.

FIGURE 5: Relative error of number of triangles under different ε . (a) Synthetic data. (b) Cora. (c) Flickr.

FIGURE 6: Relative error of normalized degree distribution under different ε . (a) Synthetic data. (b) Cora. (c) Flickr.

FIGURE 7: The modularity under different ε . (a) Synthetic data. (b) Cora. (c) Flickr.

because DP-ANPHP updates the hyperparameters during the iteration process and has a better result on clustering. Moreover, with the increase of ε , DP-ANP and DP-ANPHP have higher value of modularity. This means when the strength is small, community division has more accurate result.

7.3. Evaluation under Different K . In Figures 8–10, we show how the correlated error of average clustering coefficient, number of triangles, and normalized degree distribution vary under different K . We also compare the modularity in Figure 11. We allocate K as follows: 2, 4, 6, and 8. We can see that DP-ANP and DP-ANPHP have better performances

FIGURE 8: Relative error of average clustering coefficient under different K . (a) Synthetic data. (b) Cora. (c) Flickr.

FIGURE 9: Relative error of number of triangles under different K . (a) Synthetic data. (b) Cora. (c) Flickr.

FIGURE 10: Relative error of normalized degree distribution under different K . (a) Synthetic data. (b) Cora. (c) Flickr.

FIGURE 11: The modularity under different ϵ . (a) Synthetic data. (b) Cora. (c) Flickr.

when K is larger. The reason is that the network is divided into more groups when K is larger, and the model has more exact parameters.

8. Conclusions

In this paper, we investigate the problem of attribute network differential privacy. We use the idea of early fusion. In early fusion, it constructs a probability model combining the topological structure and the attributes together and uses it to generate a private synthetic network. We propose two attribute network differential privacy methods based on early fusion, named DP-ANP and DP-ANPHP. DP-ANP sanitizes the model parameters during the iteration process to satisfy the differential privacy. As an improvement of DP-ANP, DP-ANPHP sanitizes the model hyperparameters during the iteration process and has a better performance. The results of extensive experiments show that both DP-ANP and DP-ANPHP generate private synthetic attribute network with a high accuracy. In the future, we focus on the research of attributed network differential privacy with higher data utility.

Data Availability

The data used to support the findings of this study can be accessed from <http://konect.cc/networks/>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] L. Sweeney, “ k -anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.
- [2] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkatasubramanian, “ L -diversity: privacy beyond k -anonymity,” in *Proceedings of the International Conference on Data Engineering*, p. 24, Atlanta, GA, USA, April 2006.
- [3] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proceedings of 3rd Theory of Cryptography Conference*, pp. 265–284, New York, NY, USA, January 2006.
- [4] M. Hay, V. Rastogi, and G. Miklau, “Boosting the accuracy of differentially private histograms through consistency,” *Proceedings of VLDB Endowment*, vol. 3, no. 1-2, pp. 1021–1032, 2010.
- [5] K. Nissim, S. Raskhodnikova, and A. Smith, “Smooth sensitivity and sampling in private data analysis,” in *Proceedings of 39th Annual ACM Symposium on Theory of Computing*, pp. 75–84, San Diego, CA, USA, June 2007.
- [6] V. Karwa, S. Raskhodnikova, and A. Smith, “Private analysis of graph structure,” *ACM Transactions on Database Systems*, vol. 39, no. 3, 2014.
- [7] J. Zhang, G. Cormode, C. Procopiuc, D. Srivastava, and X. Xiao, “Private release of graph statistics using ladder functions,” in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 731–745, Melbourne, Australia, May 2015.
- [8] X. Cheng, S. Su, S. Xu, L. Xiong, K. Xiao, and M. Zhao, “A two-phase algorithm for differentially private frequent subgraph mining,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 8, pp. 1411–1425, 2018.
- [9] X. Ding, X. Zhang, Z. Bao, and H. Jin, “Privacy-preserving triangle counting in large graphs,” in *Proceedings of the Conference on Information and Knowledge Management*, pp. 1283–1292, Torino, Italy, October 2018.
- [10] H. Sun, X. Xiao, and I. Khalil, “Analyzing subgraph statistics from extended local views with decentralized differential privacy,” in *Proceedings of the 2019 ACM SIGSAC Conference*, pp. 703–717, London, UK, November 2019.
- [11] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Zhao, “Sharing graphs using differentially private graph models,” in *Proceedings of the Internet Measurement Conference*, pp. 81–98, Berlin, Germany, November 2011.
- [12] D. J. Mir and R. N. Wright, “A differentially private estimator for the stochastic Kronecker graph model,” in *Proceedings of the EDBT/ICDT Workshops*, pp. 167–176, Berlin, Germany, March 2012.
- [13] Q. Xiao, R. Chen, and K. Tan, “Differentially private network data release via structural inference,” in *Proceedings of the 20th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, pp. 911–920, New York, NY, USA, August 2014.
- [14] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren, “Generating synthetic decentralized social graphs with local

- differential privacy,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 425–438, Dallas TX USA, October 2017.
- [15] X. Chen, S. Mauw, and Y. Ramirezcruz, “Publishing community-preserving attributed social graphs with a differential privacy guarantee,” 2019, <https://arxiv.org/abs/1909.00280>.
 - [16] Y. Wang, J. Yang, and J. Zhang, “Differential privacy for weighted network based on probability model,” *IEEE Access*, vol. 8, pp. 80792–80800, 2020.
 - [17] J. Qian, X. Li, C. Zhang, and L. Chen, “De-anonymizing social networks and inferring private attributes using knowledge graphs,” in *Proceedings of the IEEE International Conference Computer and Communications*, pp. 1–9, San Francisco, CA, USA, April 2016.
 - [18] S. Ji, T. Wang, J. Chen et al., “De-SAG: on the de-anonymization of structure-attribute graph data,” *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 594–607, 2019.
 - [19] Z. Jorgensen, T. Yu, and G. Cormode, “Publishing attributed social graphs with formal privacy guarantees,” in *Proceedings of the 2016 International Conference on Management of Data*, pp. 107–122, San Francisco, CA, USA, June 2016.
 - [20] D. Yin, Y. Shen, and C. Liu, “Attribute couplet attacks and privacy preservation in social networks,” *IEEE Access*, vol. 5, pp. 25295–25305, 2017.
 - [21] M. Kiranmayi, N. Maheswari, and M. Sivagami, “Preservation of attribute couplet attack by node addition in social networks,” in *Proceedings of the International Conference on Computing and Communications Technologies*, Chennai, India, February 2019.
 - [22] R. Chen, Q. Xiao, Y. Zhang, and J. Xu, “Differentially private high-dimensional data publication via sampling-based inference,” in *Proceeding KDD 15 Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 129–138, Sydney, Australia, August 2015.
 - [23] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, “PrivBayes: private data release via bayesian networks,” *ACM Transactions on Database Systems*, vol. 42, no. 4, pp. 1–41, 2014.
 - [24] X. Ren, C.-M. Yu, W. Yu et al., “ LoPub : high-dimensional crowdsourced data publication with local differential privacy,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.
 - [25] M. Balcan, A. Blum, J. D. Hartline, and Y. Mansour, “Mechanism design via machine learning,” in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pp. 605–614, Pittsburgh, PA, USA, October 2005.
 - [26] F. Mcsherry, “Privacy integrated queries,” *Communications of the ACM*, vol. 53, no. 9, pp. 89–97, 2010.
 - [27] N. Li, W. Qardaji, and D. Su, “On sampling, anonymization, and differential privacy or, k -anonymization meets differential privacy,” in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 32–33, Seoul Korea, May 2012.