Research Article

An Edge Correlation Based Differentially Private Network Data Release Method

Junling Lu,1,2 Zhipeng Cai,3 Xiaoming Wang,1,2 Lichen Zhang,1,2 and Zhuojun Duan3

1 Key Laboratory for Modern Teaching Technology, Ministry of Education, Xi’an 710062, China
2 School of Computer Science, Shaanxi Normal University, Xi’an 710019, China
3 Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

Correspondence should be addressed to Zhipeng Cai; zcai@gsu.edu

Received 16 August 2017; Accepted 16 October 2017; Published 13 November 2017

Academic Editor: Houbing Song

Copyright © 2017 Junling Lu 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.

Differential privacy (DP) provides a rigorous and provable privacy guarantee and assumes adversaries’ arbitrary background knowledge, which makes it distinct from prior work in privacy preserving. However, DP cannot achieve claimed privacy guarantees over datasets with correlated tuples. Aiming to protect whether two individuals have a close relationship in a correlated dataset corresponding to a weighted network, we propose a differentially private network data release method, based on edge correlation, to gain the tradeoff between privacy and utility. Specifically, we first extracted the Edge Profile (PF) of an edge from a graph, which is transformed from a raw correlated dataset. Then, edge correlation is defined based on the PFs of both edges via Jenson-Shannon Divergence (JS-Divergence). Secondly, we transform a raw weighted dataset into an indicated dataset by adopting a weight threshold, to satisfy specific real need and decrease query sensitivity. Furthermore, we propose \( \varepsilon \)-correlated edge differential privacy (CEDP), by combining the correlation analysis and the correlated parameter with traditional DP. Finally, we propose network data release (NDR) algorithm based on the \( \varepsilon \)-CEDP model and discuss its privacy and utility. Extensive experiments over real and synthetic network datasets show the proposed releasing method provides better utilities while maintaining privacy guarantee.

1. Introduction

Recently, social networking such as cooperation networks, online/mobile social networks, and software defined vehicular network [1] is becoming increasingly prevalent. Accompanied with the growth of the networks, mass of network data is released for analytical decisions or scientific researches. However, direct publication of these data, including sensitive information, leads to privacy leakage of individuals. For example, whether two individuals in a social network have a close relationship may be expected to be kept a secret. Therefore, privacy concerns have been raised in increasingly emerging technologies [2–9].

In general, a dataset corresponding to such a network, usually modeled as a graph, is considered as correlated data; that is, tuples in this dataset are dependent. Clearly, privacy preserving in such correlated settings is more difficult because an adversary can infer the relationship of two individuals from their associated friends. Accordingly, our concern is preventing, whether the relationship of two individuals appears in a network dataset, from being unveiled.

Differential privacy (DP), a privacy preserving model originated from statistical database, has currently drawn considerable attentions in research communities [10–17] due to (i) its rigorous and provable privacy guarantee and (ii) its assumption of adversaries’ arbitrary background knowledge. However, DP actually assumes that the tuples in databases are independent [18]. In other words, DP cannot provide claimed privacy guarantees over correlated (nonindependent) data [19]. Therefore, the application of DP over correlated data is a challenge, and how to achieve a differentially private correlated data release method deserves to be further explored.

The focus of our work is on hiding the affinity degree of two individuals in a correlated dataset corresponding to a weighted network, that is, protecting whether the affinity degree of two individuals exceeds a given weight threshold, in a differentially private manner. Toward this end, we first transform a weighted network dataset into a corresponding
weighted graph and define the correlation of both edges via Jenson-Shannon Divergence (JS-Divergence). For satisfying specific query need in spite of some utility loss, we utilize Threshold Based Transformation (TBT) algorithm to transform a weighted dataset, adopting a weight threshold, into an indicated dataset, which also decreases query sensitivity. Finally, we present the notion of ε-Correlated Edge Differential Privacy (CEDP), by combining the correlation analysis and the correlated parameter, that is, the maximal number of correlated tuples, with traditional DP, and design differentially private network data release (NDR) algorithm to obtain better utilities while maintaining DP guarantee. Experimental results over real and synthetic network datasets also show the advantages of the proposed method. The framework of our solution is shown in Figure 1.

The contributions of our work are as follows.

First, we extract the Edge Profile (PF) vectors of edges in a weighted graph corresponding to a network dataset and then define the correlation of both edges via JS-Divergence. The inferred correlation analysis is more reasonable for datasets corresponding to such networks, since the typical Pearson correlation coefficient assumes that sample data follows normal distribution; however, the degree and weight distributions in such networks are often not so.

Second, we propose the ε-CEDP model based on our result of correlation analysis and the introduction of the correlated parameter, which makes DP over correlated datasets applicable and flexible. Furthermore, the NDR algorithm, based on correlated sensitivity and Laplace mechanism, is proposed, which also satisfies ε-CEDP and achieves the tradeoff of privacy and utility.

Third, we utilize TBT algorithm to transform a raw weighted dataset into an indicated dataset; that is, a weight value is equal to 1 or 0, by adopting a weight threshold, to satisfy specific real need and decrease query sensitivity. Admittedly, some utility loss exists in such transformation. However, many queries in real world only need Boolean values indicated by one and zero instead of accurate numeric answers. Therefore, this solution provides a feasible way for decreasing query sensitivity while maintaining real query need.

The rest of this paper is organized as follows. Section 2 discusses related literature. Section 3 provides the preliminaries. In Section 4, correlation analysis of both edges in a weighted graph is presented, and the ε-CEDP model and sensitivity calculation are proposed. Furthermore, a differentially private NDR algorithm, including TBT algorithm, to obtain the tradeoff between privacy and utility over correlated data is proposed in Section 5. The extensive experiments are illustrated in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

Compared with previous works in privacy preserving, DP proposed by Dwork [20] provides a probabilistic formulation, which represents that adversaries learn little from both databases differing in one tuple even if adversaries know about all tuples except the target one. In other words, the inference abilities of adversaries about the presence or absence of a tuple are bounded regardless of adversaries’ knowledge; that is, the presence or absence of a tuple is probabilistically indistinguishable for adversaries.

Currently, DP has drawn much attention in privacy preserving work needed in many fields. Wang et al. [10] considered a unified privacy distortion framework, where the distortion is defined to be the expected Hamming distance between the input and output databases, and investigated the relation between three different notions of privacy: identifiability, differential privacy, and mutual-information privacy. To provide personalized recommendation in big data resulting from social networks and maintain user privacy, a cloud-assisted differentially private video recommendation system based on distributed online learning was proposed [11]. The work in [12] proposed a new privacy preserving smart metering scheme for smart grid, which supports data aggregation, differential privacy, fault tolerance, and range-based filtering simultaneously. To et al. [13] introduced a novel privacy-aware framework for spatial crowdsourcing, which enables the participation of workers without compromising their location privacy. Focusing on the privacy protection of sensitive information in body area networks, the authors in [14, 15] proposed different privacy preserving schemes, based on differential privacy model, via a tree structure and dynamic noise thresholds, respectively. The work in [16] proposed a novel differentially private frequent sequence mining algorithm by leveraging a sampling-based candidate pruning technique, which satisfies ε-differential privacy and can privately find frequent sequences with high accuracy. In order to protect users’ privacy in ridesharing services, a jointly differentially private scheduling protocol has been proposed [17], which aims to protect riders’ location information and minimize the total additional vehicle mileage in the ridesharing system.

However, existing works have found that DP provides weaker privacy guarantee over nonindependent data; that is, DP needs more noise added to the output query result to cancel out the impact of correlations among tuples on privacy guarantee. Undoubtedly, how to analyze correlations among tuples and apply them into DP are desired to be further explored. For example, Kifer and Machanavajjhala [19] first
explicitly doubted the privacy guarantee of DP in correlated settings, for example, social networks, and then adopted the subsequently proposed privacy framework, that is, Pufferfish, to formalize and prove that DP assumes independence between tuples [18]. Inspired by the Pufferfish framework, Blowfish privacy [21] was proposed to achieve the tradeoff between privacy and utility using policies specifying secrets and constraints. Similarly, the authors in [22] proposed Bayesian DP to evaluate the level of private information leakage even when data is correlated and prior knowledge is incomplete. The work in [23] regarded the correlation among tuples as complete correlation and multiplied the query sensitivity with the number of correlated tuples in publishing correlated network data, which leaves room for fine-grained correlation analysis in the following work. Aiming to decrease the noise amount, Zhu et al. [24] depicted the correlation between tuples via Pearson correlation coefficient, including complete correlation, partial correlation, and independence. Liu et al. [25] inferred the dependence coefficient, distributed in interval [0, 1], to evaluate the probabilistic correlation between two tuples in a more fine-grained manner, thus reducing the query sensitivity which results in less noise. Considering temporal correlations of a moving user’s locations, the work in [26] leveraged a hidden Markov model to establish a location set and proposed a variant of DP to protect location privacy. Wu et al. [27] proposed the definition of correlated differential privacy to evaluate the real privacy level of a single dataset influenced by the other datasets when multiple datasets are correlated. The work in [28] formalized the privacy preservation problem to an optimization problem by modeling the temporal correlations among contexts and further proposed an efficient context-aware privacy preserving algorithm. Cao et al. [29] modeled the temporal correlations using Markov model and investigated the privacy leakage of a traditional DP mechanism under temporal correlations in the context of continuous data release. The work in [30] quantified the location correlation between two users through the similarity measurement of two hidden Markov models and applied differential privacy via private candidate sets to achieve the multiuser location correlation protection.

As seen from the above discussions, correlation analysis plays an important role in privacy preserving mechanisms, which directly influences the tradeoff between privacy protection and service utility. Obviously, the more accurate the correlation analysis, the better the balance of both aspects. Therefore, we attribute the underestimated privacy guarantee of DP over correlated data to the lack of data knowledge, and our work starts from data correlation analysis.

In this paper, we focus on correlated datasets corresponding to weighted cooperation networks. Different from the existing methods of correlation analysis, for example, simple multiplication in [23], Pearson correlation coefficient in [24], and the maximal information coefficient in [31], we extract the PF vectors of edges in a weighted graph corresponding to a correlated dataset and then define the correlation of both edges via JS-Divergence, which is more accurate and reasonable. Specifically, the work in [23] assumes both tuples are completely correlated; however, our proposed correlation results lie in interval [0, 1] representing multiple correlation including complete correlation. In addition, the work in [24] assumes sample data follows normal distribution, while our method is not the case. Also, the maximal information coefficient proposed in [31] satisfies two heuristic properties including generality and equitability, and we will consider it in our future work.

3. Preliminaries

3.1. Differential Privacy. Differential privacy provides the privacy guarantee for an individual in the probabilistic sense [20]. It is defined as follows.

Definition 1 (ε-differential privacy). A randomized mechanism $\mathcal{A}$ satisfies ε-differential privacy if, for any pair of databases $D$ and $D'$ differing in only one tuple and for any output $S \in O(\mathcal{A})$ representing the possible output set of $\mathcal{A}$,

$$\Pr[\mathcal{A}(D) = S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{A}(D') = S],$$

where $\varepsilon$ is the privacy budget depicting the probabilistic difference between the same outputs of $\mathcal{A}$ over $D$ and $D'$.

Generally, DP is achieved via two mechanisms: Laplace mechanism [32] and exponential mechanism [33]. Both mechanisms include a concept of global sensitivity [20], which reveals DP’s preferable choice of protecting the extreme case.

Definition 2 (global sensitivity). For any query function $f : D \rightarrow \mathbb{R}^d$, where $D$ is a dataset and $\mathbb{R}^d$ is a $d$-dimension real-valued vector, the global sensitivity of $f$ is defined as

$$\Delta f = \max_{D, D'} \|f(D) - f(D')\|_1,$$

where $D$ and $D'$ denote any pair of databases differing in only one tuple and $\| \cdot \|_1$ denotes $l_1$ norm.

Laplace mechanism, used in this paper, is formally presented as follows.

Theorem 3 (Laplace mechanism). Given any query function $f : D \rightarrow \mathbb{R}^d$, where $D$ is a dataset and $\mathbb{R}^d$ is a $d$-dimension real-valued vector, the global sensitivity $\Delta f$ of $f$, and privacy budget $\varepsilon$, a randomized mechanism $\mathcal{A}$

$$\mathcal{A}(D) = f(D) + \text{Laplace} \left( \frac{\Delta f}{\varepsilon} \right)$$

provides the ε-differential privacy, where Laplace() denotes Laplace noise.

3.2. Weighted Adjacency Matrix. In this paper, we model a correlated dataset as a weighted undirected simple graph $G = (V, E, W)$, where $V = \{v_1, \ldots, v_n\}$ is the set of vertices and $n = |V|$ is the number of vertices, $E = \{e_{ij}\}$ is the set of edges and $e_{ij} = (v_i, v_j), v_i, v_j \in V$, $i = 1, \ldots, n, v_j, v_j \in V$, $j = 1, \ldots, n$, and $W = \{w_{ij}\}$ is the set of weights where weight $w_{ij}$ corresponds...
Table 1: A raw weighted dataset.

<table>
<thead>
<tr>
<th>( V_i )</th>
<th>( V_j )</th>
<th>( w_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

Note. \( V_i \) and \( V_j \) denote two individuals, and weight \( w_{ij} \) denotes the relation strength between them.

with edge \( e_{ij} \). Then, the weighted adjacency matrix \( A^w \) of \( G \) can be denoted as

\[
A^w_{ij} = \begin{cases} 
  w_{ij}, & e_{ij} \in E \\
  0, & \text{otherwise}, 
\end{cases} \tag{4}
\]

where \( w_{ij} \) represents the affinity degree between two individuals. Obviously, the weighted adjacency matrix \( A^w_{ij} \) is symmetric.

Example 4. Suppose a raw weighted dataset \( D^w \) is listed in Table 1. Then, the corresponding weighted adjacency matrix \( A^w \) of \( D^w \) can be denoted as

\[
\begin{pmatrix}
  0 & 2 & 0 & 0 & 0 & 0 \\
  2 & 0 & 4 & 8 & 1 & 0 \\
  0 & 4 & 0 & 0 & 0 & 0 \\
  0 & 8 & 0 & 0 & 5 & 3 \\
  0 & 1 & 0 & 5 & 0 & 0 \\
  0 & 0 & 0 & 3 & 0 & 0 \\
\end{pmatrix}. \tag{5}
\]

3.3. Correlation Metric. Motivated by the entropy in information theory, we adopt JS-Divergence, inferred from Kullback-Leibler Divergence (KL-Divergence) \cite{23}, to depict the difference of two distributions, which can be transformed to depict the correlation of two tuples in a correlated dataset.

Definition 5 (KL-Divergence). Suppose \( P = \{p_1, p_2, \ldots, p_n\} \) and \( Q = \{q_1, q_2, \ldots, q_n\} \) are the probability distributions of random variables \( X = \{x_1, x_2, \ldots, x_n\} \) and \( Y = \{y_1, y_2, \ldots, y_n\} \), then the KL-Divergence of \( P \) and \( Q \) is defined as follows:

\[
KLD (P \parallel Q) = \sum_{i=1}^{n} p_i \ln \left( \frac{p_i}{q_i} \right). \tag{6}
\]

Here \( \log 0 = 0 \) is required. Based on KL-Divergence, we can obtain JS-Divergence as follows.

Definition 6 (JS-Divergence). Suppose \( P = \{p_1, p_2, \ldots, p_n\} \) and \( Q = \{q_1, q_2, \ldots, q_n\} \) are the probability distributions of random variables \( X = \{x_1, x_2, \ldots, x_n\} \) and \( Y = \{y_1, y_2, \ldots, y_n\} \), and \( M = (1/2)(P + Q) \); then the JS-Divergence of \( P \) and \( Q \) is defined as follows:

\[
JSD (P \parallel Q) = \frac{1}{2} (KLD (P \parallel M) + KLD (Q \parallel M))
\]

\[= \frac{1}{2} \sum_{i=1}^{n} \left( p_i \ln \frac{2p_i}{p_i + q_i} + q_i \ln \frac{2q_i}{p_i + q_i} \right). \tag{7}
\]

4. Correlation Analysis of Weighted Edges

In this section, we first discuss how to define the correlation of both edges in a weighted graph corresponding to a network dataset and then introduce why and how we conduct dataset transformation based on a given weight threshold. Finally, we define the \( e \)-CEDP model and calculate the correlated sensitivity for smaller added noise.

4.1. Correlation Definition. For achieving the correlation of tuples in a raw weighted dataset \( D^w \), we first obtain a weighted graph \( G \), whose weighted adjacency matrix is denoted by \( A^w \). Then, the correlation problem is changed to seeking the correlation of edges in \( G \). To this end, we first describe the PF vector of a weighted edge from the perspectives of relational strength and network structure and then define the correlation of both edges via JS-Divergence instead of Pearson correlation coefficient.

For a weighted edge \( e_{ij} \), suppose \( V_i = \{v \mid (v, v_i) \in E, v \neq v_i\} \) represents the set of vertices connected with \( v_i \) and \( V_j = \{v \mid (v, v_j) \in E, v \neq v_j\} \) represents the set of vertices connected with \( v_j \); we extract the PF vector of \( e_{ij} \), denoted by \( PF(e_{ij}) \), from the perspectives of relational strength and network structure simultaneously. Specifically, we obtain \( w_{ij}/\max_{v \in V_i} w_{ij} \in [0, 1] \) from the global weights of all edges. In addition, we get \( w_{ij}/\sum_{v \in V_i} w_{ij} \in [0, 1] \) and \( w_{ij}/\sum_{v \in V_j} w_{ij} \in [0, 1] \) from the local weights of edge \( e_{ij} \). On the other hand, similar to the representation of relational strength, \( \text{Deg}(v_i)/\max_{v \in V} \text{Deg}(v) \in [0, 1] \) and \( \text{Deg}(v_j)/\max_{v \in V} \text{Deg}(v) \in [0, 1] \) are constructed, by introducing the node degree \( \text{Deg}(v) \), to depict the global active degree for both vertices of edge \( e_{ij} \). Also, \( |V_i \cap V_j|/|V_i \cup V_j| \in [0, 1] \) is adopted, via the set similarity, to depict the ratio of the number of common vertices connecting \( v_i \) and \( v_j \) to that of \( V_i \) and \( V_j \). Meanwhile, \( \text{Deg}(v_i)/\sum_{v \in V_i} \text{Deg}(v) \in [0, 1] \) and \( \text{Deg}(v_j)/\sum_{v \in V_j} \text{Deg}(v) \in [0, 1] \) are used to depict the local active degree for both vertices of edge \( e_{ij} \). Combining the above factors, we obtain \( PF(e_{ij}) \) as follows:

\[
PF(e_{ij}) = \left( \frac{w_{ij}}{\max_{v \in V_i} w_{ij}}, \frac{w_{ij}}{\sum_{v \in V_i} w_{ij}}, \frac{w_{ij}}{\sum_{v \in V_j} w_{ij}}, \frac{\text{Deg}(v_i)}{\max_{v \in V} \text{Deg}(v)}, \frac{\text{Deg}(v_j)}{\sum_{v \in V} \text{Deg}(v)} \right). \tag{8}
\]
Similarly, for any other edge $e_{mn}$, according to (8), we define $PF(e_{mn})$ as follows:

$$PF(e_{mn}) = \left( \frac{w_{mn}}{\max_{e \in E} (w_e)} \right)^{\frac{\sum_{v \in V_n} w_{mv}}{\sum_{v \in V_n} w_{nv}}} \cdot \frac{\deg(v_m)}{\max_{v \in V} \deg(v)} \cdot \frac{\deg(v_n)}{\max_{v \in V} \deg(v)} \cdot \frac{|V_m \cap V_n|}{|V_m \cup V_n|}. \quad (9)$$

Note that Pearson correlation coefficient assumes that sample data follows normal distribution. However, in social networks, the weight and degree distributions do not follow such distribution, which is also verified by our experiments shown in Figure 2, where geom and out are the abbreviations of geom.net [34–36] and out.moreno_lesmis_lesmis [37–39] for simplicity. Specifically, (i) geom.net is the authors collaboration network in Computational Geometry based on the file geombib.bib, and the reduced simple network contains 7343 vertices and 11898 edges. Two authors are linked with an edge, iff they wrote a common work. The value of an edge is the number of common works. (ii) out.moreno_lesmis_lesmis is the characters cooccurrences network in Victor Hugo’s novel “Les Misérables,” and it contains 77 vertices and 254 edges. A node represents a character and an edge between two nodes shows that these two characters appeared in the same chapter of the book. The weight of each link indicates how often such a coappearance occurred.

Since our constructed PF vectors of edges do not satisfy the assumption of normal distribution, we adopt JS-Divergence, instead of Pearson correlation coefficient, to measure the CORrelation (COR) of any two edges $e_{ij}$, $e_{mn}$ in a weighted graph. To this end, we normalize $PF(e_{ij})$ and...
where $S_{ij} = w_{ij}/\max_{e \in E} (w_{ij}) + w_{ij}/\sum_{v \in V} w_{ij} + \text{Deg}(v_i)/\max_{e \in V} \text{Deg}(v) + \text{Deg}(v_i)/\sum_{v \in V} \text{Deg}(v)$ and

$$\text{PN}(e_{mn}) = \frac{\text{PF}(e_{mn})}{S_{mn}}, \quad (11)$$

where $S_{mn} = w_{mn}/\max_{e \in E} (w_{mn}) + w_{mn}/\sum_{v \in V} w_{mn} + w_{mn}/\sum_{v \in V} w_{mn} + \text{Deg}(v_m)/\max_{e \in V} \text{Deg}(v) + \text{Deg}(v_m)/\sum_{v \in V} \text{Deg}(v) + \text{Deg}(v_m)/\sum_{v \in V} \text{Deg}(v)$.

Meanwhile, we consider the distance of both edges as follows.

**Definition 7** (edge distance). Suppose $e_{ij} = (v_i, v_j)$ and $e_{mn} = (v_m, v_n)$ are two edges in graph $G$, $d(v_i, v_j)$ denotes the length of the shortest path between nodes $v_i$ and $v_j$, and $I$ is the index of the smallest value in vector $[d(v_i, v_m), d(v_i, v_n), d(v_j, v_m), d(v_j, v_n)]$; then the distance of $e_{ij}$ and $e_{mn}$ is defined as follows:

$$\text{dis}(e_{ij}, e_{mn}) = \begin{cases} 
    d(v_i, v_m) + d(v_j, v_n), & I = 1 \\
    d(v_i, v_m) + d(v_j, v_m), & I = 2 \\
    d(v_i, v_n) + d(v_j, v_m), & I = 3 \\
    d(v_i, v_n) + d(v_j, v_n), & I = 4.
\end{cases} \quad (12)$$

Specifically, we first calculate the distances including $d(v_i, v_m), d(v_i, v_n), d(v_j, v_m), d(v_j, v_n)$ and then determine the smallest one among these distances and its index $I$, complete the calculation of the distance of another pair of nodes, and finally obtain the distance of the two edges $e_{ij}$ and $e_{mn}$.

Based on Definitions 6 and 7, we define the CORrelation (COR) of two probability distributions via JS-Divergence as follows.

**Definition 8** (CORrelation). Suppose $P = \{p_1, p_2, \ldots, p_k\}$ and $Q = \{q_1, q_2, \ldots, q_k\}$ are the probability distributions of random variables $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$; then the CORrelation of $P$ and $Q$ is defined as follows:

$$\text{COR}(P, Q) = \frac{1 - \text{JSD}(P \parallel Q)}{1 + \text{dis}(e_{ij}, e_{mn})}, \quad (13)$$

According to (10)–(13) and Definition 6, we adopt the normalized PN vectors of edges $e_{ij}$, $e_{mn}$ to measure their correlation as

$$\text{COR}(e_{ij}, e_{mn}) = \frac{1}{1 + \text{dis}(e_{ij}, e_{mn})}, \quad (14)$$

$$\text{COR}(e_{ij}, e_{mn}) = \frac{1}{1 + \text{dis}(e_{ij}, e_{mn})}, \quad (15)$$

Substituting (15) into (14), we obtain

$$\text{COR}(e_{ij}, e_{mn}) = \frac{1}{1 + \text{dis}(e_{ij}, e_{mn})}, \quad (16)$$

In our opinion, the proposed correlation definition, extracted from two aspects of relational strength and network structure, is more reasonable. The rationale is (i) graph models, commonly abstracted from networks, reflect inherent dependent relations of individuals, which naturally form edge correlations and (ii) the weights of edges in weighted graphs describe the affinity degree of individuals’ relations, which also influence the variances of edge correlations.

**Example 9.** Take $e_{24}$ and $e_{25}$ in $D_W$ as an example to demonstrate the calculation of COR($e_{24}$, $e_{25}$). According to (8) and (9), we have

$$\text{PF}(e_{24}) = \left(\frac{8}{7}, \frac{8}{15}, \frac{8}{16}, \frac{4}{7}, \frac{4}{7}, \frac{7}{6}, \frac{7}{7}\right), \quad (17)$$

$$\text{PF}(e_{25}) = \left(\frac{1}{8}, \frac{1}{15}, \frac{6}{4}, \frac{4}{7}, \frac{5}{7}, \frac{7}{7}\right).$$

Furthermore, according to (10) and (11), we get

$$\text{PN}(e_{24}) = (0.2020, 0.1077, 0.1010, 0.2020, 0.1515, 0.0337, 0.1154, 0.0866), \quad (18)$$

$$\text{PN}(e_{25}) = (0.0429, 0.0229, 0.0572, 0.3430, 0.1715, 0.0686, 0.1960, 0.0980). \quad (19)$$
Definition 10 (correlated neighboring databases). Any pair of databases $D_{\text{COR}_z}$ and $D'_{\text{COR}_z}$ are correlated neighboring databases, if the weight change of a tuple in $D_{\text{COR}_z}$ results in the weight changes of at most $z - 1$ other correlated tuples in $D'$ based on the correlation $\text{COR}(\cdot, \cdot)$ of both tuples.

Note that the neighboring databases in Definition 10 are described by two parameters: the correlation $\text{COR}(\cdot, \cdot)$ aforementioned and the correlated parameter $z$. Specifically, we have the following:

(i) Based on JS-Divergence, we have the following conclusion about the correlation $\text{COR}(\cdot, \cdot)$.

\textbf{Theorem 11.} For any two edges $e_{ij}$ and $e_{mn}$ in the weighted graph $G$ corresponding to a network dataset $D_W$, $0 \leq \text{COR}(e_{ij}, e_{mn}) \leq 1$ holds.

\textit{Proof.} For the ease of exposition, we denote the last two items in the numerator of (16) as follows.

$$\text{JSD}(e_{ij} \parallel e_{mn}) = \frac{1}{2} \cdot \sum_{k=1}^{8} \left( \frac{\text{PN}(e_{ij})_k \ln \frac{2\text{PN}(e_{ij})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k}}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k} \right) + \frac{\text{PN}(e_{mn})_k \ln \frac{2\text{PN}(e_{mn})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k}}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k}.$$  

Since $\text{PN}(\cdot)$ denotes a probability distribution, we have

$$\sum_{k=1}^{8} \text{PN}(e_{ij})_k = 1,$$

$$\sum_{k=1}^{8} \text{PN}(e_{mn})_k = 1;$$

we consider two cases separately.

\textbf{Case 1} ($\text{COR}(e_{ij}, e_{mn}) \leq 1$). Since $\ln(x) \leq x - 1$ when $x > 0$, we have

$$\sum_{k=1}^{8} \text{PN}(e_{ij})_k \ln \frac{2\text{PN}(e_{ij})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k} \leq \sum_{k=1}^{8} \text{PN}(e_{ij})_k \left( \frac{2\text{PN}(e_{ij})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k} - 1 \right)$$

$$= \sum_{k=1}^{8} \text{PN}(e_{ij})_k \frac{\text{PN}(e_{ij})_k - \text{PN}(e_{mn})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k}$$

$$\leq \sum_{k=1}^{8} \text{PN}(e_{ij})_k.$$ 

Substituting (22) into (23), we have

$$\sum_{k=1}^{8} \text{PN}(e_{ij})_k \ln \frac{2\text{PN}(e_{ij})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k} \leq 1.$$ 

\textbf{Case 2} ($\text{COR}(e_{ij}, e_{mn}) > 1$). Similarly, we obtain

$$\sum_{k=1}^{8} \text{PN}(e_{mn})_k \ln \frac{2\text{PN}(e_{mn})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k} \leq 1.$$
Combining (21), (24), and (25), we have
\[ \text{JSD} \left( e_{ij} \parallel e_{mn} \right) \leq 1. \quad (26) \]

**Case 2** (\( \text{COR}(e_{ij}, e_{mn}) \geq 0 \)). Since \( \ln(x) \geq 1 - 1/x \) when \( x > 0 \), we have
\[
\sum_{k=1}^{8} \frac{\text{PN}(e_{ij})_k \ln \frac{2\text{PN}(e_{ij})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k}}{\text{PN}(e_{ij})_k}
\geq \sum_{k=1}^{8} \ln \left( 1 - \frac{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k}{2\text{PN}(e_{ij})_k} \right)
\]
\[
= \sum_{k=1}^{8} \frac{\text{PN}(e_{ij})_k - \text{PN}(e_{mn})_k}{2}. \quad (27)
\]

Similarly, we obtain
\[
\sum_{k=1}^{8} \frac{\text{PN}(e_{mn})_k \ln \frac{2\text{PN}(e_{mn})_k}{\text{PN}(e_{ij})_k + \text{PN}(e_{mn})_k}}{\text{PN}(e_{ij})_k}
\geq \sum_{k=1}^{8} \frac{\text{PN}(e_{mn})_k - \text{PN}(e_{ij})_k}{2}. \quad (28)
\]

Combining (21), (27), and (28), we have
\[ \text{JSD} \left( e_{ij} \parallel e_{mn} \right) \geq 0. \quad (29) \]

Combining (26) with (29), then we have
\[ 0 \leq \text{JSD} \left( e_{ij} \parallel e_{mn} \right) \leq 1. \quad (30) \]

Note that \( \text{dis}(e_{ij}, e_{mn}) \geq 0 \) due to (12). Finally, according to (16) and (30), we have
\[ 0 \leq \text{COR} \left( e_{ij}, e_{mn} \right) \leq 1. \quad (31) \]

Clearly, \( \text{COR}(e_{ij}, e_{mn}) = 0 \) denotes \( e_{ij} \) is independent of \( e_{mn} \); that is, the corresponding tuples in a dataset are independent. \( \text{COR}(e_{ij}, e_{mn}) = 1 \) denotes \( e_{ij} \) is fully dependent to \( e_{mn} \); that is, the corresponding tuples in a dataset are fully correlated. \( 0 < \text{COR}(e_{ij}, e_{mn}) < 1 \) denotes \( e_{ij} \) is partially dependent on \( e_{mn} \); that is, the corresponding tuples in a dataset are partially correlated.

(ii) Similar to [23–25], we introduce the correlated parameter \( z \) representing that there are at most \( z \) correlated tuples in a dataset. In other words, a tuple is correlated with at most \( z - 1 \) other tuples; that is, an edge in a graph is correlated with at most \( z - 1 \) other edges. Obviously, \( z = 1 \) represents the independent case of tuples in a dataset, \( z = n \) represents the fully correlated case of tuples in a dataset, and \( 1 < z < n \) represents the partially correlated case of tuples in a dataset. Therefore, the variance of \( z \) increases the flexibility of Definition 10.

Furthermore, we define the \( \epsilon \)-CEDP model as follows.

**Definition 12** (\( \epsilon \)-correlated edge differential privacy). A randomized mechanism \( \mathcal{M} \) satisfies \( \epsilon \)-differential privacy if, for any neighboring databases \( D_{\text{COR},z} \) and \( D'_{\text{COR},z} \) and for any output \( S \in O(\mathcal{M}) \) representing the possible output set of \( \mathcal{M} \),
\[
\Pr(\mathcal{M}(D_{\text{COR},z}) = S) \leq \exp(\epsilon) \cdot \Pr(\mathcal{M}(D'_{\text{COR},z}) = S), \quad (32)
\]
where \( \epsilon \) is the privacy budget depicting the probabilistic difference between the same outputs of \( \mathcal{M} \) over \( D_{\text{COR},z} \) and \( D'_{\text{COR},z} \) and \( \text{COR}(\cdot, \cdot) \) and \( z \) are the correlation of two tuples and the correlated parameter representing the maximal number of correlated tuples, respectively.

**4.4. Sensitivity Calculation.** After transforming weighted dataset \( D_W \) to indicated dataset \( D_I \), we add Laplace noise to query answers based on the \( \epsilon \)-CEDP model. Laplace noise is determined by two factors: privacy budget \( \epsilon \) and the global sensitivity of a query, and the latter refers to the maximal change of query result due to the modification of only one tuple. Here, for a query \( f \), assume the global sensitivity of \( f \), resulting from the change of tuple \( t_j \), in independent settings is \( \Delta f_j \). Clearly, \( \Delta f_j = 1 \). However, for dataset \( D_I \) with \( n \) tuples where at most \( z \) tuples are correlated, the query sensitivity resulted from modifying tuple \( t_j \); called Edge Sensitivity denoted by \( S_{E_j} \) is more complex. Specifically, (i) if \( \text{COR}(t_i, t_j) = 0 \), that is, \( z = 1 \), denoting the independent case, \( S_{E_j} = 1 \), (ii) if \( \text{COR}(t_i, t_j) = 1 \) and \( 2 \leq z \leq n \), denoting the fully correlated case, \( S_{E_j} = z \), and (iii) if \( 0 < \text{COR}(e_{ij}, e_{mn}) < 1 \) and \( 2 \leq z \leq n \), denoting the partially correlated case, \( S_{E_j} \) is defined as follows.

\[ S_{E_j} = \sum_{j=1}^{n} \left| \text{COR} \left( t_i, t_j \right) \right| \Delta f_j \quad (33) \]

Since the change of a tuple only affects at most other \( z - 1 \) correlated tuples, \( S_{E_j} \) can be rewritten as
\[ S_{E_j} = \sum_{j=1}^{z} \left| \text{COR} \left( t_i, t_j \right) \right| \Delta f_j. \quad (34) \]

Finally, we have the correlated sensitivity denoted by \( S_{C} \), that is, the maximal \( S_{E_j} \) in dataset \( D_I \), as follows:
\[ S_{C} = \max_{i \in D_I} S_{E_j}. \quad (35) \]

Note that the \( S_{C} \) is also suitable for the independent and fully correlated cases. Based on the \( S_{C} \), we can achieve \( \epsilon \)-CEDP, which is shown as follows.

**Theorem 13.** Given any query function \( f : D_{\text{COR},z} \rightarrow \mathbb{R}^d \), where \( D_{\text{COR},z} \) is a correlated dataset with the correlation definition \( \text{COR}(\cdot, \cdot) \) and the correlated parameter \( z \) and \( \mathbb{R}^d \) is a \( d \)-dimension real-valued vector, the correlated sensitivity \( S_{C} \) of \( f \), and privacy budget \( \epsilon \), a randomized mechanism \( \mathcal{M} \),
\[
\mathcal{M}(D_{\text{COR},z}) = f(D_{\text{COR},z}) + \text{Laplace} \left( \frac{S_{C}}{\epsilon} \right), \quad (36)
\]
provides \( \epsilon \)-CEDP, where \( \text{Laplace}(\cdot) \) denotes Laplace noise.
the tradeoff between privacy and utility under correlated
5.1. NDR Algorithm.
Section 4, we proposed a network data release method in
privacy budget ε, correlated parameter z, threshold T and query set ℓ.
Output: Noisy query result ℳ(DG).
(1) Calculate the correlation COR(⋅, ⋅) of any two edges in DW according to Eq. (16);
(2) Call Algorithm TBT(DW, T), return D1;
(3) Calculate the correlated sensitivity CS according to Eq. (35);
(4) for (Each f ∈ ℓ) do
(5) ℳ(Df) = f(Df) + Laplace(\frac{CS}{ε});
(6) end for
(7) return Noisy query result ℳ(DG) as ℳ(DW).

Algorithm 2: NDR algorithm.

Proof.
\[
\frac{Pr(ℳ(D_{COR}, z) = S)}{Pr(ℳ(D'_{COR}, z) = S)} = \frac{\exp(-\varepsilon |ℳ(D_{COR}, z) - f(D_{COR}, z)|/CS)}{\exp(-\varepsilon |ℳ(D'_{COR}, z) - f(D'_{COR}, z)|/CS)} \leq \exp\left(\frac{T \delta}{CS}\right).
\]

According to (35), the following holds:
\[
\left|\frac{f(D_{COR}, z) - f(D'_{COR}, z)}{CS}\right| \leq 1.
\]

Finally, combining (37) with (38), we have
\[
\frac{Pr(ℳ(D_{COR}, z) = S)}{Pr(ℳ(D'_{COR}, z) = S)} \leq \exp(\varepsilon).
\]

For indicated dataset D1 with weight w_{ij} ∈ {0, 1} and the correlated parameter z, we can easily infer the global sensitivity is equal to z. Due to 0 ≤ COR(⋅, ⋅) ≤ 1, we have CS < z. Therefore, CS is less than the global sensitivity. In other words, added noise via CS is less than that via the global sensitivity; hence the utility of the mechanism ℳ based on CS is better.

5. Network Data Release Method

Based on indicated dataset D1 and the CS discussed in Section 4, we proposed a network data release method in special cases, which achieves the ε-CEDP model. Furthermore, the theoretical analysis of privacy and utility is elaborated.

5.1. NDR Algorithm. The goal of NDR algorithm is to achieve the tradeoff between privacy and utility under correlated settings. To this end, three phases are taken into account: (i) for achieving the correlation of two tuples in DW, we transform dataset DW into the corresponding graph and calculate the correlation COR(⋅, ⋅) of both edges via the JS-Divergence, (ii) based on a given weight threshold T, we convert DW into D1 via TBT algorithm. In other words, the sensitivity in independent settings is 1, irrelevant to the weights. Furthermore, we implement the calculation of CS, and (iii) combining the affordable privacy budget ε with CS, we calculate the added Laplace noise and finally obtain the noisy query result ℳ(Df) for query f in query set ℓ. The NDR algorithm is presented in Algorithm 2.

5.2. Utility Analysis. Clearly, NDR algorithm satisfies the ε-CEDP model. To conduct utility analysis, we adopt the (α, δ)-useful definition in [40] to depict the utility of NDR as follows.

Definition 14. A mechanism NDR is (α, δ)-useful for a query f in all queries ℓ, if, with probability at least 1 − δ, for any query f and dataset D, NDR satisfies max_{f∈ℓ}|NDR(D) − f(D)| ≤ α.

Based on Definition 14, we obtain the following utility analysis.

Theorem 15. For any query f ∈ ℓ and dataset D, a mechanism NDR satisfies (α, δ)-useful if NDR can obtain max_{f∈ℓ}|NDR(D) − f(D)| ≤ α with at least probability 1 − δ when δ ≥ \exp(-εα/CS).

Proof. By Definition 14, we have
\[
\frac{Pr\left(\max_{f∈ℓ}|NDR(D) − f(D)| \leq α\right)}{Pr\left(\max_{f∈ℓ}\left|\text{Laplace}\left(\frac{CS}{ε}\right)\right| \leq α\right)} = \frac{\int_{-\alpha}^{\alpha} \frac{ε}{2CS} \exp\left(-\frac{εx}{CS}\right) dx}{\int_{0}^{\alpha} \frac{ε}{CS} \exp\left(-\frac{εx}{CS}\right) dx} = 1 - \exp\left(-\frac{εα}{CS}\right).
\]

If δ ≥ \exp(-εα/CS), then the following holds:
\[
1 - \exp\left(-\frac{εα}{CS}\right) \leq 1 - δ.
\]
According to (40) and (41), we obtain
\[ \Pr \left( \max_{f \in \mathcal{Q}} |NDR(D) - f(D)| \leq \alpha \right) \leq 1 - \delta. \] (42)

Therefore, mechanism NDR satisfies \((\alpha, \delta)\)-useful.

6. Experiment

Generally, the goal of privacy preserving is to achieve maximal utilities while maintaining required privacy guarantees; that is, the tradeoff between privacy and utility is desired. In this section, we first present the better privacy guarantees and utilities of Algorithm NDR based on the definition of \((\alpha, \delta)\)-useful and then further demonstrate its better utilities in terms of mean absolute error (MAE). Here the Baseline algorithm adopts the multiplication in [23] to handle with the correlated tuples in a network dataset. Considering the constraint of applying Pearson correlation coefficient, we do not adopt the method using Pearson correlation coefficient as comparison reference in the following experiments. To verify the advantages of Algorithm NDR concerning privacy and utility, we conduct NDR and Baseline algorithms on three datasets: geom, out explained in Section 4, and randomly generated dataset (rgd), which is a randomly generated weighted network containing 100 vertices and 1645 edges. The weight of each edge is uniformly distributed in interval [1, 50]. Such doing can also show the better adaption of the proposed correlation metric and algorithm over real world and synthetic datasets. Without loss of generality, threshold \( T \) here is set as 0, and the selection of its value is to be investigated in future work.

6.1. Privacy and Utility. We analyzed privacy and utility of NDR and Baseline algorithms in terms of \((\alpha, \delta)\)-useful when the correlated parameter is set to the size of the whole dataset. In terms of privacy, we evaluate the consumption of privacy budget \( \epsilon \) under the same accuracy \( \alpha \) and the same possibility \( 1 - \delta \). Clearly, the smaller the consumed privacy budget, the better the performance of algorithm.

Figures 3(a), 3(c), and 3(e) present the variation of privacy budget, consumed by algorithms NDR and Baseline based on datasets geom, out, and rgd, with the increase of \( \alpha \) from 1 to 40 when \( \delta \) equals 0.1 and 0.5, respectively. From Figures 3(a), 3(c), and 3(e), we can see that privacy budgets decrease in all cases with the increase of \( \alpha \). The reason is that, with the relaxation of \( \alpha \), larger noise can be allowed when \( \delta \) stays fixed; therefore algorithm can consume smaller privacy budget. Meanwhile, we also see that privacy budgets decrease with the increase of \( \delta \) from 0.1 to 0.5 when \( \alpha \) stays fixed. Because the possibility of satisfying accuracy requirement decreases with the increase of \( \delta \), which means that algorithms can have more chances to add larger noise; that is, algorithms can use smaller privacy budget. Such advantage of both algorithms especially in the case of \( \alpha = 1 \) is more obvious than that of other ones. In fact, when \( \delta \) stays constant, the higher the accuracy presented by \( \alpha \), the more the privacy budget needed by algorithms.

On the other hand, Figures 3(b), 3(d), and 3(f) demonstrate the variation of \( \delta \) of algorithms NDR and Baseline based on datasets geom, out, and rgd, with the increase of \( \alpha \) from 0 to 10000 when \( \epsilon \) equals 0.1 and 1.0, respectively. We can see that \( \delta \) decreases in all cases with the increase of \( \alpha \); that is, the possibility increases with the increase of \( \alpha \). Note that this trend varies from dataset to dataset; for example, the possibility and accuracy of algorithm NDR over datasets geom and out are evidently different when \( \epsilon = 1 \). Clearly, when \( \epsilon \) is determined, the possibility increases with the relaxation of \( \alpha \). In addition, we find that algorithm NDR can have larger possibility, that is, smaller \( \delta \), than the Baseline algorithm to achieve the same accuracy \( \alpha \) under the same level of privacy budget \( \epsilon \). Also, when \( \epsilon \) increases from 0.1 to 1.0, algorithms also have possibility to achieve the same accuracy \( \alpha \), which is easily understood from (40).

6.2. Utility. We adopt MAE, that is, \((1/|\mathcal{Q}|) \sum_{f \in \mathcal{Q}} |\hat{f}(t) - f(t)|\), to depict the performance of algorithms NDR and Baseline. Obviously, the smaller the MAE value, the better the utility. For each dataset, 10000 queries are randomly generated, and each query result ranges from 0 to the maximal number of tuples.

Figures 4(a), 4(c), and 4(e) show the variances of MAEs of NDR and Baseline algorithms, over datasets geom, out, and rgd, under various privacy budgets when the correlated parameter \( z \) is 10. From Figures 4(a), 4(c), and 4(e), we can see that the MAEs of both algorithms decrease with the increase of privacy budget \( \epsilon \) from 0.1 to 1. Because larger privacy budget leads to smaller noise added to raw data, the downtrends always hold. More importantly, algorithm NDR can obtain better accuracy, that is, smaller MAE, under various \( \epsilon \). Furthermore, the smaller the privacy budget \( \epsilon \), the more obvious such advantage. The reason is that algorithm NDR adopts the more reasonable correlation metric compared with the Baseline algorithm.

Figures 4(b), 4(d), and 4(f) show the variances of MAEs of NDR and Baseline algorithms, over datasets geom, out, and rgd, under various correlated parameters when privacy budget \( \epsilon \) is 0.5. In Figures 4(b), 4(d), and 4(f), we find that the MAEs of both algorithms increase with the increase of correlated parameter \( z \) from 1 to 40. Undoubtedly, with the increase of the number of correlated tuples in a dataset, larger noise needs to be injected to eliminate the effect of tuple correlation, which necessarily results in the increase of MAE. In addition, we also note that algorithm NDR can obtain better accuracy, that is, smaller MAE, under various correlated parameters compared with the Baseline algorithm. Also, the larger the correlated parameter, the larger such advantage. All these advantages are due to the more reasonable correlation metric, which is proposed in Section 4 and adopted by algorithm NDR.

7. Conclusion

In this paper, we focus on adopting differential privacy model to avoid the leakage of close relationship between two individuals in a network. To this end, we first extract the PF vector from both aspects of node degree and edge weight to depict an edge in a network dataset and then design the correlation metric of two edges via JS-Divergence to avoid the
Figure 3: Comparison of privacy and utility on network data in terms of $(\alpha, \delta)$-useful.
Figure 4: Comparison of utility on network data in terms of mean absolute error.
constraint of adopting Pearson correlation coefficient. Next, we proposed the $\epsilon$-CEDP model to deal with the correlated dataset by introducing two parameters including our correlation metric and the correlated parameter. Furthermore, we present the NDR algorithm based on the $\epsilon$-CEDP and discuss its privacy and utility in terms of the definition of $(\alpha, \delta)$-useful. Extensive experiments on real and synthetic network datasets verify the advantages of our proposed privacy preserving model and algorithm concerning privacy and utility. Admittedly, the proposed solution is currently appropriate for weighted network datasets, and other datasets are out of the scope of this paper. In future work, we will discuss the impacts of choosing weight threshold on algorithm performances, explore more appropriate correlation metrics, and investigate privacy preserving algorithms in different applications.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partly supported by the Fundamental Research Funds for the Central Universities of China under Grant no. GK201703061, the National Natural Science Foundation of China under Grants nos. 61402273 and 61373083, the National Science Foundation (NSF) under Grants nos. CNS-1252292, 1741277, and 1704287, and the Natural Science Basic Research Plan in Shaanxi Province of China under Grants nos. 2017JM6060 and 2017JM6103.

References


