LEARNING PRIVACY-PRESERVING GRAPH EMBEDDING AGAINST SENSITIVE ATTRIBUTE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We focus on preserving the privacy of some sensitive attributes associated with certain private nodes on a graph when releasing graph data. Notably, deleting the sensitive attributes from the graph data cannot resist adversarial attacks because an adversary can still leverage the graph structure information and the non-sensitive node features to predict the sensitive attributes. We propose a framework to learn graph embeddings insensitive to the changes of certain specified sensitive attributes while maximally preserving the graph structure information and non-sensitive node features for downstream tasks. The key ingredient of our framework is a novel conditional variational graph autoencoder (CVGAE), which captures the relationship between the learned embeddings and the sensitive attributes. This allows us to quantify the privacy loss that can be used for penalizing privacy leakage when learning graph embeddings without adversarial training.

1 INTRODUCTION

As the world becomes more connected, data generated from individuals typically are not independent but exhibit inherent correlations. As a consequence, individual information that is seemingly innocent in online social networks can be used to infer sensitive attributes of targeted individuals (Gong & Liu, 2016; 2018), which has led to a growing concern among the general public about privacy breaches. Meanwhile, graph neural networks (GNNs) (Zhou et al., 2020) were developed for learning graph-structured data and have achieved success in various domains such as product recommendation (Ying et al., 2018) and knowledge base completion (Hamaguchi et al., 2017). The significant progress of GNNs concomitantly exacerbates the privacy problem because GNNs can also be powerful tools for an adversary to learn sensitive attributes (Sun et al., 2022) without direct access to them. Consider user profile data collected by social media where some users choose to hide their ethnicity while others do not. The data is shared with a third party for an authorized task. Masking the ethnicity of those concerned users cannot prevent an untrusted third party from inferring the masked attributes. This is partly because of the homophily property that users with similar attributes tend to form closer connections compared to dissimilar ones (McPherson et al., 2001). In addition, sensitive attributes can be correlated with other non-sensitive features. For example, ethnic groups may have their own hobby preferences. Therefore, an adversary can capitalize on the social network links and the public user features to estimate the masked sensitive attributes (cf. graph-based missing data prediction (You et al., 2020)). What is worse is that the leaked sensitive attribute may be further exploited to discriminate and make biased decisions against users. The General Data Protection Regulation, which came into effect in Europe in 2018, legally requires organizations to ensure appropriate security and confidentiality when handling data. Therefore, it becomes imperative for organizations to sanitize graph data to protect sensitive attributes before sharing the data.

Popular privacy-preserving approaches such as k-anonymity (Sweeney, 2002), l-diversity (Machanavajjhala et al., 2006) and t-closeness (Rebollo-Monedero et al., 2010) are specifically created for tabular data and do not take into account the topology or correlation structure of graph-structured data. Therefore, these approaches do not prevent the leakage of sensitive information caused by the connections among linked nodes. They do not simultaneously consider the privacy leakages from node features and graph structures. The statistical privacy models based on information-theoretic perspectives can defend against adversarial statistical inference (Calmon & Fawaz, 2012; Sankar et al., 2013). These, however, require explicit knowledge of the underlying data distribution

and are difficult to generalize to graph data where the joint distribution of node features and graph edges is usually implicit.

GNNs have proven successful in learning informative node representations in homophilic graphs (Hamilton et al., 2017a;b) through feature propagation and aggregation. The learned representation can be used for various downstream tasks ranging from node classification to link prediction (Kipf & Welling, 2017). The latent variables learned by GNNs reside in a regular Euclidean space, which enables us to perform data sanitization without having to attend to the irregular graph domain.

In this paper, we propose privacy-preserving encoder and decoder architectures to learn latent representations of graph-structured data for which specified sensitive attributes on private nodes are obfuscated. At the same time, we conserve as much of the useful information about the graph structure and non-sensitive node features as possible. The learned graph representations can be shared with a third party for it to perform other downstream tasks. In particular, we consider two scenarios, depending on whether or not the graph structure is available to the adversarial third party as ancillary information.

Proposed method. Our method is inspired by the variational graph autoencoder (VGAE) (Kipf & Welling, 2016), which is a framework for unsupervised learning on graph-structured data based on the variational autoencoder (VAE) (Kingma & Welling, 2014). However, the VGAE can inherit sensitive information from training data, making the latent representations of graph data vulnerable to inference attacks when being accessed by an adversary. We propose conditional variational graph autoencoder (CVGAE), in which we model the dependence of the marginal distribution of the latent variable on an input (such as the sensitive attributes) as a parameterized Gaussian channel. Based on this stochastic relationship between the sensitive attributes and the latent variable, we construct a penalty for privacy leakage and apply the CVGAE to encourage it to disentangle the graph representations from the sensitive attributes. To further mitigate privacy leakage, we add Gaussian noise to the learned graph representations to provide differential privacy (Dwork et al., 2006; Dwork & Roth, 2014).

2 Related Work

The problem of preserving inference privacy is analogous to the task of factoring out undesired variations in representation learning. In this regard, learning invariant or fair representations has been well studied by the machine learning community. Works like Louizos et al. (2016); Moyer et al. (2018) proposed encoder and decoder architectures to learn latent representations that are invariant to certain known variation factors. However, these works are premised on the availability of independently and identically distributed (i.i.d.) data, and thus not directly applicable to graph-structured data.

Addressing inference privacy in graph-structured data domains is still in its infancy stage. An empirical approach to this problem is using adversarial training (Li et al., 2021; Zhang & Zitnik, 2020). This method traces back to Huang et al. (2018), where finding the optimal sanitization mechanism is formulated as a competing game between a sanitizer and an adversary. However, adversarial training is known to be unstable and the quality of privacy sanitization is determined by the capability of the chosen adversarial neural network, which in practice cannot incorporate all possible adversarial strategies. This implies that such approaches are not guaranteed to achieve a universal privacy protection level. In this paper, instead of using an adversarial network, we instead explicitly incorporate a privacy leakage penalty in our learning architecture.

It is worth highlighting that there is increasing interest in the issue of fairness in graph-structured data learning (Agarwal et al., 2021; Dai & Wang, 2022; Fan et al., 2021; Wang et al., 2022). The fairness task aims to correct model bias by enforcing statistical parity (such as demographic parity) of a specific task with with respect to (w.r.t.) certain sensitive factors. This formulation differs from the privacy protection task considered in this paper. In the privacy protection task, the target is to purge sensitive attributes from the representations and hence to limit the capability of an adversary in inferring sensitive attributes from the representations (Agarwal, 2021). Moreover, many of the fairness works are based on adversarial training, which thus suffer from the above-mentioned non-universality.

The most relevant work to this paper is Hu et al. (2022), in which the authors seek to disentangle the node features into sensitive and non-sensitive latent representations by imposing orthogonality (in a suitable space). However, orthogonal elements can still be correlated statistically, leading to privacy leakage. Our method outperforms this approach, as demonstrated in the experiments.

3 PROBLEM FORMULATION

In this section, we formalize the problem of protecting sensitive attributes of private nodes in a homogeneous graph (where nodes and edges are of the same types).

We are given an undirected and unweighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$ nodes. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be the adjacency matrix of $\mathcal{G}$, where $\mathbf{A}_{i,j} \neq 0$ in the ith row and jth column indicates an edge between nodes i and j. Let $\mathbf{D}$ be the degree matrix. A matrix

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\intercal \in \mathbb{R}^{N \times D},$$

where $\mathbf{x}_i$ represents the ith node feature vector, summarizes the public or non-private node features associated with each node.

Sensitive attributes and private nodes. Each node is associated with a sensitive attribute, which is assumed to take values in a set $\mathbb{S} \subset \mathbb{R}$ for simplicity. The sensitive attributes of all nodes are collected as

$$\mathbf{s} = [\mathrm{s}_1, \ldots, \mathrm{s}_N]^\intercal .$$

The set of nodes $\mathcal{V}$ is partitioned into two groups, indexed by index sets $\mathcal{P}$ and $\mathcal{Q}$, respectively. Nodes indexed by $\mathcal{P}$ are called private nodes, which do not want to reveal their sensitive attributes. The nodes indexed by $\mathcal{Q}$ are called public nodes, which are nonchalant about exposing their sensitive attributes. See Fig. 1 for an illustration.

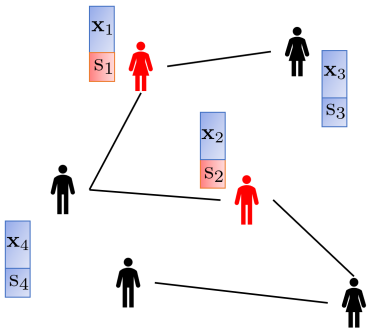

Figure 1: Social networks where private users (in red color) have sensitive attributes (in red color).

Suppose the graph data owner wants to share the graph data $(\mathbf{A}, \mathbf{X})$ with a third-party organization to perform downstream tasks like node classification. As alluded to in the first paragraph in Section 1, simply hiding the sensitive attributes of the private nodes cannot stop the third party from gaining information about the sensitive attributes $\mathbf{s}_\mathcal{P} \coloneqq \{\mathrm{s}_i : i \in \mathcal{P}\}$ due to the following two reasons. Firstly, an adversary may collect $\mathbf{s}_\mathcal{Q} \coloneqq \{\mathrm{s}_i : i \in \mathcal{Q}\}$ from the public nodes in advance and make use of it as side information to predict $\mathbf{s}_\mathcal{P}$. Secondly, $\mathbf{s}_\mathcal{P}$ can be correlated with $\mathbf{X}$.

Our task is to learn graph representations $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^\intercal \in \mathbb{R}^{N \times M}$ from $(\mathbf{A}, \mathbf{X}, \mathbf{s})$ that can prevent $\mathbf{s}_\mathcal{P}$ from being inferred by an adversary. We consider the following two cases:

(a) The adversary has a priori knowledge of both $\mathbf{s}_\mathcal{Q}$ and the graph topology $\mathbf{A}$.

(b) The adversary knows $\mathbf{s}_\mathcal{Q}$ but not $\mathbf{A}$.

For both cases (a) and (b), the goal is to learn graph representations that do not significantly increase the level of the adversary's information about $\mathbf{s}_\mathcal{P}$. At the same time, we attempt to maximally retain information about graph structure and node features so that the utility of other downstream tasks is not significantly affected. To achieve this, we propose encoder and decoder architectures in Section 4 to learn a graph embedding $\mathbf{Z}$ dependent on $\mathbf{s}$. Then we discuss privacy sanitization schemes in Section 5.

4 LEARNING EMBEDDINGS

In this section, we present our conditional variational graph autoencoder (CVGAE) to learn latent graph embeddings that can be used for downstream tasks. The proposed CVGAE characterizes the probabilistic channel $p(\mathbf{Z} \mid \mathbf{A}, \mathbf{s})$ and $p(\mathbf{Z} \mid \mathbf{s})$ (where the latent graph embedding $\mathbf{Z}$ depends on the input $\mathbf{s}$). This allows us to form privacy penalty functions for case (a) (where $\mathbf{A}$ is a priori knowledge) and case (b), respectively, in Section 5.

It is not straightforward to generalize the conditional variational autoencoder (CVAE) (Sohn et al., 2015) designed for statistical data to graph-structured data. This is because CVAE requires i.i.d. data samples to train parameterized models to learn the probabilistic channel between the latent variable and a variable of interest. However, the single-shot graph data sample $(\mathbf{A}, \mathbf{X}, \mathbf{s})$ makes it infeasible to learn this probabilistic channel using the stochastic method. To solve this, we introduce parameterized models that can traverse the graph nodes to capture the high-level node features. This can be simply done with the graph convolutional network (GCN) and multilayer perceptron (MLP). For example, a one-layer GCN and MLP can be written as

$$\mathrm{GCN}(\mathbf{A}, \mathbf{s}) = \mathrm{ReLU}(\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{s}\mathbf{w}^{\intercal}), \tag{1}$$

$$\mathrm{MLP}(\mathbf{s}) = \mathrm{ReLU}(\mathbf{s}\mathbf{w}^{\intercal}), \tag{2}$$

where the trainable weights $\mathbf{w}$ are shared by every element in $\mathbf{s}$. Therefore, as long as $\mathbf{s}$ has a sufficient number of elements, we are able to learn the domain knowledge of $\mathbf{s}$ (since the domain of each s in $\mathbf{s}$ is the same due to the homogeneous graph assumption).

4.1 ADVERSARY HAS TOPOLOGY INFORMATION

We present CVGAE assuming that the adversary has prior knowledge of $\mathbf{A}$. Note that since $\mathbf{A}$ is prior knowledge, we are not necessarily encoding the graph structure into the latent representation $\mathbf{Z}$. From the factorization

$$p(\mathbf{X} \mid \mathbf{A}, \mathbf{s}) = \frac{p(\mathbf{X} \mid \mathbf{A}, \mathbf{Z}, \mathbf{s}) p(\mathbf{Z} \mid \mathbf{A}, \mathbf{s})}{p(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}, \mathbf{s})},$$

we obtain the evidence lower bound (ELBO) of $p(\mathbf{X} \mid \mathbf{A}, \mathbf{s})$ as

$$\mathbb{E}_{q(\mathbf{Z}|\mathbf{A},\mathbf{X},\mathbf{s})}[\log p(\mathbf{X} \mid \mathbf{A}, \mathbf{Z}, \mathbf{s})] - D_{\mathrm{KL}}(q(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}, \mathbf{s}) \,\|\, p(\mathbf{Z} \mid \mathbf{A}, \mathbf{s})), \tag{3}$$

in which $q(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}, \mathbf{s})$ is the variational encoder, $p(\mathbf{X} \mid \mathbf{A}, \mathbf{Z}, \mathbf{s})$ is the decoder, $p(\mathbf{Z} \mid \mathbf{A}, \mathbf{s})$ is the marginal distribution of $\mathbf{Z}$ conditioned on $(\mathbf{A}, \mathbf{s})$, and $D_{\mathrm{KL}}(\cdot \,\|\, \cdot)$ denotes the Kullback-Leibler divergence.

We let the encoder and decoder be parameterized by multi-layer GCNs:

$$q(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}, \mathbf{s}) = \prod_{i=1}^{N} q(\mathbf{z}_i \mid \mathbf{A}, \mathbf{X}, \mathbf{s}), \text{ with } q(\mathbf{z}_i \mid \mathbf{A}, \mathbf{X}, \mathbf{s}) = \mathcal{N}\left(\mathbf{z}_i \,\middle|\, \boldsymbol{\mu}_i, \mathrm{diag}(\boldsymbol{\sigma}_i^2)\right),$$

$$p(\mathbf{X} \mid \mathbf{A}, \mathbf{Z}, \mathbf{s}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \mathbf{A}, \mathbf{Z}, \mathbf{s}), \text{ with } p(\mathbf{x}_i \mid \mathbf{A}, \mathbf{Z}, \mathbf{s}) = \mathcal{N}\left(\mathbf{x}_i \,|\, \boldsymbol{\phi}_i, \mathbf{I}_D\right),$$

where $\mathrm{diag}(\mathbf{a})$ denotes the diagonal matrix with diagonal entries given by $\mathbf{a}$, $\mathbf{I}_D$ is the identity matrix of size $D \times D$ and $\mathcal{N}(\mathbf{z}_i \,|\, \boldsymbol{\mu}_i, \mathbf{I}_D)$ denotes the normal distribution over $\mathbf{z}_i$ with mean $\boldsymbol{\mu}_i$ and covariance $\mathbf{I}_D$; $[\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N]$, $[\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_N]$ and $[\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N]$ are the outputs from GCNs (i.e., multi-layer variants of (1)) denoted as $\mathrm{GCN}_{\boldsymbol{\mu}}(\mathbf{A}, [\mathbf{X}, \mathbf{s}])$, $\mathrm{GCN}_{\boldsymbol{\sigma}}(\mathbf{A}, [\mathbf{X}, \mathbf{s}])$ and $\mathrm{GCN}_{\boldsymbol{\phi}}(\mathbf{A}, [\mathbf{Z}, \mathbf{s}])$, respectively. Let the conditional marginal distribution be

$$p(\mathbf{Z} \mid \mathbf{A}, \mathbf{s}) = \prod_{i=1}^{N} p(\mathbf{z}_i \mid \mathbf{A}, \mathbf{s}), \text{ with } p(\mathbf{z}_i \mid \mathbf{A}, \mathbf{s}) = \mathcal{N}\left(\mathbf{z}_i \,|\, \boldsymbol{\nu}_i, \mathbf{I}_M\right),$$

where $[\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_N]$ are the outputs of $\mathrm{GCN}_{\boldsymbol{\nu}}(\mathbf{A}, \mathbf{s})$. Fig. 2 illustrates this model.

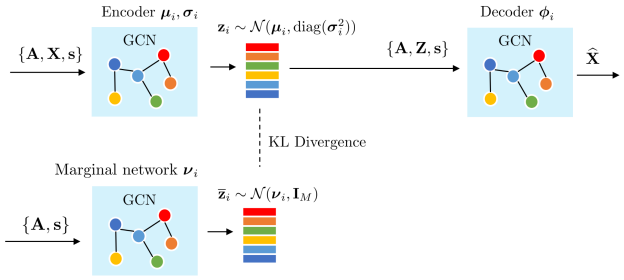

Figure 2: CVGAE assuming adversary has knowledge of $\mathbf{A}$.

4.2 Adversary has no topology information

We now present CVGAE assuming that the adversary does not know the graph topology. Since the graph adjacency matrix $\mathbf{A}$ is not known a priori, we encode the node features as well as the graph structure into $\mathbf{Z}$. From

$$p(\mathbf{A}, \mathbf{X} \mid \mathbf{s}) = \frac{p(\mathbf{X}, \mathbf{A} \mid \mathbf{Z}, \mathbf{s}) p(\mathbf{Z} \mid \mathbf{s})}{p(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}, \mathbf{s})},$$

and assuming $p(\mathbf{X}, \mathbf{A} \mid \mathbf{Z}, \mathbf{s}) = p(\mathbf{X} \mid \mathbf{Z}, \mathbf{s}) p(\mathbf{A} \mid \mathbf{Z}, \mathbf{s})$, we obtain the ELBO of $p(\mathbf{A}, \mathbf{X} \mid \mathbf{s})$ as

$$\mathbb{E}_{q(\mathbf{Z}|\mathbf{A},\mathbf{X},\mathbf{s})}[\log p(\mathbf{X} \mid \mathbf{Z}, \mathbf{s})] + \mathbb{E}_{q(\mathbf{Z}|\mathbf{A},\mathbf{X},\mathbf{s})}[\log p(\mathbf{A} \mid \mathbf{Z}, \mathbf{s})] - D_{\mathrm{KL}}(q(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}, \mathbf{s}) \,\|\, p(\mathbf{Z} \mid \mathbf{s})), \tag{4}$$

where $q(\mathbf{Z} \mid \mathbf{A}, \mathbf{X}, \mathbf{s})$ is the variational encoder, $p(\mathbf{X} \mid \mathbf{Z}, \mathbf{s})$ and $p(\mathbf{A} \mid \mathbf{Z}, \mathbf{s})$ are the decoder of node features and the decoder of adjacency matrix, respectively, and $p(\mathbf{Z} \mid \mathbf{s})$ is the marginal distribution of $\mathbf{Z}$ conditioned on $\mathbf{s}$.

We parameterize the decoders as

$$p(\mathbf{A} \mid \mathbf{Z}, \mathbf{s}) = \prod_{i=1}^{N} \prod_{j=1}^{N} p(\mathbf{A}_{i,j} \mid \mathbf{z}'_i, \mathbf{z}'_j), \text{ with } p(\mathbf{A}_{i,j} \mid \mathbf{z}'_i, \mathbf{z}'_j) = \sigma(\boldsymbol{\psi}(\mathbf{z}'_i)^{\intercal} \boldsymbol{\psi}(\mathbf{z}'_j)),$$

$$p(\mathbf{X} \mid \mathbf{Z}, \mathbf{s}) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \mathbf{z}'_i), \text{ with } p(\mathbf{x}_i \mid \mathbf{z}'_i) = \mathcal{N}\left(\mathbf{x}_i \mid \boldsymbol{\phi}(\mathbf{z}'_i), \mathbf{I}_D\right),$$

where $\mathbf{z}'_i = [\mathbf{z}_i^{\intercal}, \mathrm{s}_i]^{\intercal}$, $\sigma(\cdot)$ is the Sigmoid function, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are trainable MLPs. The conditional marginal distribution is modeled as

$$p(\mathbf{Z} \mid \mathbf{s}) = \prod_{i=1}^{N} p(\mathbf{z}_i \mid \mathrm{s}_i) \text{ where } p(\mathbf{z}_i \mid \mathrm{s}_i) = \mathcal{N}\left(\mathbf{z}_i \mid \boldsymbol{\nu}(\mathrm{s}_i), \mathbf{I}_M\right),$$

where $\boldsymbol{\nu}$ is a trainable MLP. This model is illustrated in Fig. 3.

We make use of the reparameterization trick (Kingma & Welling, 2014) and apply gradient descent to train the CVGAEs.

5 Privacy Sanitization

In this section, we discuss data sanitization schemes to maximally purge the sensitive attributes $\mathbf{s}_{\mathcal{P}}$ of the private nodes from the latent graph embedding $\mathbf{Z}$ when training the CVGAEs. In addition, we ensure controllable differential privacy (DP) w.r.t. a set of privacy candidates by post-processing the learned graph embedding.

Adversary model. Recall the assumption in Section 3 that the adversary has collected all the sensitive attributes of the public nodes $\mathbf{s}_{\mathcal{Q}}$ beforehand (since the public nodes do not mind disclosing these

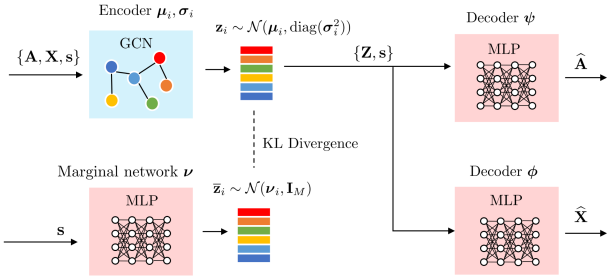

Figure 3: CVGAE assuming adversary has no knowledge of $\mathbf{A}$

attributes). When the adversary has access to the released graph embedding $\mathbf{Z}$, it can estimate the sensitive attributes $\mathbf{s}_\mathcal{P}$ by taking advantage of the side information $\mathbf{A}$ and $\mathbf{s}_\mathcal{Q}$ for case (a) or $\mathbf{s}_\mathcal{Q}$ for case (b). In both scenarios, the strategy adopted by the adversary to infer $\mathbf{s}_\mathcal{P}$ is unlikely to be known to the data owner in practice. We address the privacy provision from a DP viewpoint.

To obfuscate the information of $\mathbf{s}$ in the latent graph embedding $\mathbf{Z}$, we need to define a set of privacy candidates who share the same data type as the sensitive attribute $\mathbf{s}$. Let $\boldsymbol{s}$ be the observation of $\mathbf{s}$.

Definition 1 (Privacy candidates)**.** *A set $\mathcal{S}$ is said to contain privacy candidates of $\boldsymbol{s} = [s_1, \ldots, s_N]^\intercal$ if $|\mathcal{S}| > 1$ and for any $\boldsymbol{s}' = [s'_1, \ldots, s'_N]^\intercal \in \mathcal{S}$, $s'_i = s_i$, $\forall\, i \in \mathcal{Q}$ and $s'_i \in \mathbb{S}$, $\forall\, i \in \mathcal{P}$.*

Basically, the sensitive attributes of the public nodes are the same for all the elements in $\mathcal{S}$. This is because $\boldsymbol{s}_\mathcal{Q}$ is assumed to be known to the adversary. Any prediction $\boldsymbol{s}'$ such that $\boldsymbol{s}'_\mathcal{Q} \neq \boldsymbol{s}_\mathcal{Q}$ would be immediately rejected by the adversary due to $p(\mathbf{s} = \boldsymbol{s}' \mid \mathbf{s}_\mathcal{Q} = \boldsymbol{s}_\mathcal{Q}) = 0$. However, the definition of privacy candidates can still be pathological for case (a) because the adversary also has prior knowledge of $\mathbf{A}$. For $\boldsymbol{s}' \in \mathcal{S}$ such that $p(\mathbf{A}, \mathbf{s} = \boldsymbol{s}') < p(\mathbf{A}, \mathbf{s} = \boldsymbol{s})$, the adversary is likely to reject $\boldsymbol{s}'$, which increases the probability of a correct guess of $\mathbf{s}_\mathcal{P}$. Therefore, including more elements in $\mathcal{S}$ does not necessarily enhance privacy in case (b).

5.1 Differential privacy

DP (Dwork et al., 2006) has been deemed a gold-standard within the privacy community. It requires that the output of an enquiry on a database should not differ much if we arbitrarily perturb the database by only one data point. However, the original definition of DP does not fit into the scope of our work, which aims to protect the sensitive attributes. In what follows, we tailor DP to meet our needs. Suppose we have chosen the privacy candidates $\mathcal{S}$ of $\boldsymbol{s}$.

Definition 2 (Differential privacy)**.** *Let $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$. For a graph embedding $\mathbf{Z}$, we say $\mathbf{Z}$ achieves (ϵ, δ)-DP w.r.t. $\mathcal{S}$ if for any $\boldsymbol{s}' \in \mathcal{S}$, we have*

$$p(\mathbf{Z} \mid \mathbf{s} = \boldsymbol{s}) = \mathrm{e}^{-\epsilon} p(\mathbf{Z} \mid \mathbf{s} = \boldsymbol{s}') + \delta, \text{ and } p(\mathbf{Z} \mid \mathbf{s} = \boldsymbol{s}') = \mathrm{e}^{-\epsilon} p(\mathbf{Z} \mid \mathbf{s} = \boldsymbol{s}) + \delta. \tag{5}$$

This ensures that with probability at least $1 - \delta$ (Dwork & Roth, 2014, Lemma 3.17), the distribution of $\mathbf{Z}$ conditioned on $\mathbf{s}$ is indistinguishable from that conditioned on any other privacy candidate. Note that we need to replace $p(\mathbf{Z} \mid \mathbf{s})$ with $p(\mathbf{Z} \mid \mathbf{s}, \mathbf{A})$ in (5) for case (a) where the adversary has prior knowledge of $\mathbf{A}$.

Gaussian mechanism (Dwork & Roth, 2014). The Gaussian mechanism adds noise drawn from a Gaussian distribution whose variance is calibrated according to the sensitivity and privacy parameters to guarantee DP. This can be restated as: two Gaussian distributions with mean difference $\Delta\boldsymbol{\mu}$ satisfy (5) if they have variance at least $\gamma(\Delta\boldsymbol{\mu})\mathbf{I}$ where

$$\gamma(\Delta\boldsymbol{\mu}) = \frac{2\log(1.25/\delta)(\|\Delta\boldsymbol{\mu}\|)^2}{\epsilon^2}, \tag{6}$$

with $\Delta\boldsymbol{\mu}$ being called the sensitivity. It can be deduced that we may either increase the variance or decrease the mean difference of the two Gaussian distributions to satisfy (5).

5.2 PRIVACY PENALTY

Recall the conditional marginal distribution in the CVGAE model for case (b) where the adversary has no knowledge of $\mathbf{A}$: $p(\mathbf{Z} \mid \mathbf{s}) = \prod_{i=1}^{N} p(\mathbf{z}_i \mid \mathrm{s}_i)$ with $p(\mathbf{z}_i \mid \mathrm{s}_i) = \mathcal{N}(\mathbf{z}_i \mid \boldsymbol{\nu}(\mathrm{s}_i), \mathbf{I}_M)$. To provide (ϵ, δ)-DP for the embedding $\mathbf{Z}$ w.r.t. $\mathcal{S}$, we append the following penalty term into the objective function of the ELBO (4) to strike a balance between utility (reconstruction loss) and privacy:

$$\Delta\boldsymbol{\nu} = \max_{\boldsymbol{s}' \in \mathcal{S}} \sum_{i=1}^{N} \|\boldsymbol{\nu}(s_i) - \boldsymbol{\nu}(s_i')\|^2.$$

This is to encourage the CVGAE to minimize the mean difference between the Gaussian distributions $p(\mathbf{Z} \mid \mathbf{s} = \boldsymbol{s})$ and $p(\mathbf{Z} \mid \mathbf{s} = \boldsymbol{s}')$ for $\boldsymbol{s}' \in \mathcal{S}$. We call the CVGAE with this privacy penalty the privacy-preserving CVGAE (PPCVGAE).

To achieve exact (ϵ, δ)-DP, we can further inject noise to the embedding:

$$\widetilde{\mathbf{z}}_i = \mathbf{z}_i + \mathbf{n}_i,$$

where $\mathbf{n}_i$ is an zero-mean independent Gaussian noise vector with variance being $\max(\gamma(\Delta\boldsymbol{\nu}) - 1, 0)$. Note that adding excessive noise to data can cause large distortion to the data. Our method reduces the amount of additive noise needed for DP. We call this approach PPCVGAE+DP.

The same strategy applies to case (a) where the adversary knows $\mathbf{A}$, by noting that the conditional marginal distribution in CVGAE in this case is $p(\mathbf{Z} \mid \mathbf{s}, \mathbf{A}) = \prod_{i=1}^{N} p(\mathbf{z}_i \mid \mathbf{A}, \mathbf{s})$ with $p(\mathbf{z}_i \mid \mathbf{A}, \mathbf{s}) = \mathcal{N}(\mathbf{z}_i \mid \boldsymbol{\nu}_i, \mathbf{I}_M)$, where $\boldsymbol{\nu}_i \coloneqq \boldsymbol{\nu}_i(\mathbf{A}, \mathbf{s})$ denotes the ith output from $\mathrm{GCN}_{\boldsymbol{\nu}}(\mathbf{A}, \mathbf{s})$. For case (a), $\Delta\boldsymbol{\nu}$ becomes

$$\Delta\boldsymbol{\nu} = \max_{\boldsymbol{s}' \in \mathcal{S}} \sum_{i=1}^{N} \|\boldsymbol{\nu}_i(\mathbf{A}, \boldsymbol{s}) - \boldsymbol{\nu}_i(\mathbf{A}, \boldsymbol{s}')\|^2.$$

It is worth highlighting that $\Delta\boldsymbol{\nu}$ bounds the mutual information between $\mathbf{Z}$ and $\mathbf{s}$ and a small value of $\Delta\boldsymbol{\nu}$ can guarantee a large detection error of $\mathbf{s}$ for an adversary. We provide the proof in Appendix A.

6 EXPERIMENTS

In this section, we conduct numerical experiments on five real-world graph datasets to demonstrate the effectiveness of our proposed privacy framework, i.e., PPCVGAE. We examine the privacy-utility trade-off by performing sensitive attribute inference and node classification on the embedding learned from PPCVGAE. We compare against DP-GCN (Hu et al., 2022) and use VGAE as as baseline to compare with the performance on unsanitized embeddings.

6.1 DATASET DESCRIPTION

The datasets used in this experiment include two social networks datasets: Pokec-z and Pokec-n (Takac & Zabovsky, 2012; Dai & Wang, 2022), and three ethical datasets: German credit, Recidivism and Credit defaulter (Agarwal et al., 2021). Information of the experiment datasets is summarized in Table 1.

- The nodes in the German credit dataset represent bank clients and the node features are client profile information such as credit amount, job and age. The edges are formed between clients based on the similarity of their credit accounts. We treat gender as the sensitive attribute and the utility task is to classify clients as having good or bad credit risks.
- The nodes in the Credit defaulter graph are credit card users and the nodes features contain information on default payments, demographic factors, credit data, history of payment and bill statements of the credit card users. Edges are formed between users if they share similar patterns in purchases and payments. The utility task is to predict whether or not an individual will default on the credit card payment while age is the sensitive attribute.
- Pokec-z and Pokec-n are anonymized social network datasets, where edges represent friendships of users and node features contain attributes like gender, age, hobbies, interest, and education. We select region as the sensitive attribute, and the utility target is to classify users' working field.

- The Recidivism graph has its nodes representing defendants who are released on bail at the U.S. state courts from 1990 to 2009. Edges are constructed based on the similarity of past criminal records and demographics. The goal is to classify defendants into two categories: "bail" (i.e., unlikely to commit a violent crime if released) versus "no bail" (i.e., likely to commit a violent crime), while race is chosen as the sensitive attribute.

Dataset	Pokec-z	Pokec-n	German credit	Recidivism	Credit defaulter
#Nodes	67796	66569	1000	18876	30000
#Edges	13033712	1100663	22242	321308	1436858
#Features	257	264	27	18	13
Sens. Attr	Region	Region	Gender	Race	Age
Label	Working Field	Working Field	Good/bad Credit	Bail/no Bail	Default/no default

Table 1: Dataset summary.

6.2 Experimental Setup

We randomly choose 70% of the nodes as the private nodes (indexed by $\mathcal{P}$) whose sensitive attributes are to be protected. The rest of the 30% nodes (indexed by $\mathcal{Q}$) are the public nodes whose sensitive attributes are available to the adversary. We let the privacy candidates be $\mathcal{S} = \{\boldsymbol{s}, \boldsymbol{s}'\}$, where $\boldsymbol{s}$ is the ground-truth sensitive labels of all the graph nodes and $\boldsymbol{s}'$ is created with $s'_i = s_i$ for $i \in \mathcal{Q}$ and s'_i taking the opposite of the ground-truth label for $i \in \mathcal{P}$.

We test our framework for case (a) where the adversary has access to $(\mathbf{A}, \mathbf{s}_\mathcal{Q})$ as side information and case (b) where the adversary has access to only $\mathbf{s}_\mathcal{Q}$ as side information, respectively. We train the PPCVGAE discussed in Section 5 to obtain privacy-preserving graph embeddings. The PPCVGAE adopts the CVGAE given inSection 4.1 for case (a), while PPCVGAE uses the CVGAE in Section 4.2 for case (b). Apart from that, we add Gaussian noise to the learned graph embeddings to achieve (ϵ, δ)-DP with $\epsilon = 0.01 = \epsilon = 10^{-4}$. This is denoted as PPCVGAE+DP.

After obtaining the sanitized graph embedding from PPCVGAE or PPCVGAE+DP, we quantify the empirical utility and privacy by performing the sensitive attribute inference and utility node classification on the sanitized embedding, respectively. For case (a), we train two separate GCNs for attribute inference on $\mathbf{s}_\mathcal{P}$ and node classification for utility by taking $(\mathbf{A}, \mathbf{Z}, \mathbf{s}_\mathcal{Q})$ as inputs. For case (b), we use two MLP classifiers for sensitive attribute inference and utility node classification by taking $(\mathbf{Z}, \mathbf{s}_\mathcal{Q})$ as inputs. For comparison purposes, we perform the utility and privacy tasks on the unsanitized embeddings learned by a normal VGAE (Kipf & Welling, 2016). Moreover, we compare our method with DP-GCN from Hu et al. (2022) for case (a). The utility results are shown in Table 2, while the privacy attack results are shown in Table 3. The optimal privacy-utility trade-offs are highlighted in bold.

We follow the data splitting and prepossessing in (Hu et al., 2022). The data is randomly partitioned into 50%/30%/20% for training, validation and testing for the utility task of transductive node classification. We also notice that the Credit defaulter dataset is highly imbalanced w.r.t. the sensitive attribute labels. Therefore, we implement resampling to balance the number of positive and negative samples in both the training and test sets.

6.3 Results and Analysis

An interesting observation from the results on German credit and Credit defaulter in Table 3 is that both PPCVGAE and PPCVGAE+DP in case (a) did not appreciably bring down the accuracy of adversarial attack (relative to the results on the unsanitized embedding). This is because the adversary in case (a) has the prior knowledge of the adjacency matrix $\mathbf{A}$ and the sensitive attributes on the public nodes $\mathbf{s}_\mathcal{Q}$. The adjacency matrices of these two datasets are artificially formed based on similarity measures. Thus $\mathbf{s}_\mathcal{Q}$ is very smooth on $\mathbf{A}$, which allows the adversary to predict the sensitive attributes purely based on the side information $(\mathbf{A}, \mathbf{s}_\mathcal{Q})$. In contrast to case (a), the adversary in case (b) only possesses $\mathbf{s}_\mathcal{Q}$ and it will need more information about the graph data to predict $\mathbf{s}_\mathcal{P}$. Therefore, we are able to reduce the accuracy of adversarial inference via the sanitized graph embedding.

Dataset		Pokec-z	Pokec-n	German credit	Recidivism	Credit defaulter
VGAE	Acc	84.78 ± 0.26	86.64 ± 0.17	67.43 ± 3.37	90.81 ± 0.26	78.44 ± 1.32
	F1	84.78 ± 0.26	86.64 ± 0.17	78.41 ± 2.81	86.81 ± 0.23	86.88 ± 0.83
DP-GCN	Acc	84.95 ± 0.26	86.73 ± 0.19	70.40 ± 3.17	82.08 ± 1.14	68.85 ± 1.31
for case (a)	F1	84.95 ± 0.26	86.73 ± 0.19	81.51 ± 3.29	73.24 ± 1.54	70.84 ± 1.08
PPCVGAE	Acc.	85.12 ± 0.27	86.39 ± 0.18	72.70 ± 3.25	**85.32** ± 0.79	75.00 ± 1.67
for case (a)	F1	85.12 ± 0.27	86.39 ± 0.18	83.25 ± 2.34	**78.40** ± 0.98	85.48 ± 1.15
PPCVGAE+DP	Acc.	79.42 ± 4.22	**77.73** ± 6.51	76.65 ± 4.32	83.83 ± 0.98	75.47 ± 1.58
for case (a)	F1	79.42 ± 4.22	**77.73** ± 6.51	84.05 ± 3.54	75.02 ± 3.12	84.56 ± 1.45
PPCVGAE	Acc.	**85.12** ± 0.27	86.44 ± 0.24	68.10 ± 2.90	75.43 ± 7.25	72.79 ± 1.66
for case (b)	F1	**85.12** ± 0.27	86.44 ± 0.24	80.99 ± 2.07	58.24 ± 22.20	85.57 ± 1.08
PPCVGAE+DP	Acc.	84.91 ± 0.22	86.73 ± 0.18	**68.10** ± 2.90	62.68 ± 0.35	**64.71** ± 17.06
for case (b)	F1	84.91 ± 0.22	86.73 ± 0.18	**80.99** ± 2.07	0.14 ± 0.20	**72.07** ± 19.82

Table 2: Utility performance. Results of node classification. Higher value indicates better utility.

Dataset		Pokec-z	Pokec-n	German credit	Recidivism	Credit defaulter
VGAE	Acc	96.61 ± 0.06	96.74 ± 0.10	95.88 ± 0.56	58.72 ± 0.53	95.88 ± 0.61
	F1	95.17 ± 0.09	94.39 ± 0.16	93.53 ± 1.09	59.90 ± 1.63	95.83 ± 0.60
DP-GCN	Acc	97.97 ± 0.08	98.17 ± 0.05	93.77 ± 1.50	63.84 ± 0.33	97.41 ± 0.31
for case (a)	F1	97.13 ± 0.11	96.84 ± 0.08	89.92 ± 2.79	63.94 ± 2.97	97.40 ± 0.31
PPCVGAE	Acc.	65.50 ± 4.14	70.83 ± 0.02	92.00 ± 1.01	**52.01** ± 0.77	87.39 ± 1.99
for case (a)	F1	4.44 ± 2.51	0.36 ± 0.15	86.94 ± 2.45	**53.83** ± 3.10	87.40 ± 2.03
PPCVGAE+DP	Acc.	50.13 ± 0.16	**51.23** ± 3.01	92.20 ± 1.45	50.86 ± 1.31	86.88 ± 3.34
for case (a)	F1	35.47 ± 35.63	**27.79** ± 25.87	92.43 ± 1.45	48.56 ± 23.17	87.87 ± 2.50
PPCVGAE	Acc.	**64.85** ± 0.16	71.00 ± 0.20	71.43 ± 4.86	55.63 ± 4.80	82.84 ± 3.33
for case (b)	F1	**0.0** ± 0.0	0.0 ± 0.0	15.09 ± 24.50	62.54 ± 5.65	83.29 ± 3.34
PPCVGAE+DP	Acc.	64.93 ± 0.19	60.31 ± 21.12	**62.17** ± 16.40	50.09 ± 0.60	**52.00** ± 2.32
for case (b)	F1	0.0 ± 0.0	11.13 ± 22.26	**8.89** ± 22.12	48.57 ± 6.51	**26.02** ± 12.27

Table 3: Privacy performance. Results of sensitive attribute inference. Lower value indicates stronger privacy.

By comparing PPCVGAE and PPCVGAE-DP for both case (a) and case (b) in Tables 2 and 3, it can be seen that DP has adverse effect on both the tasks of node classification and sensitive attributes inference. This is because the additive noise needed for DP is not calibrated for a particular task, thus distorting the overall data information. A stronger privacy guarantee is always at the cost of lower utility. However, the cost of utility varies on a case-by-case basis. For example, the privacy-utility trade-off is very decent on Pokec-z, Pokec-n and German credit. This is possibly because the sensitive attributes are less correlated with the node features useful for the utility task. On the contrary, the accuracy of the utility task for Recidivism dropped significantly when the accuracy of sensitive attribute inference is reduced to 50%.

We observe that our approaches outperform the DP-GCN method in terms of the privacy-utility tradeoff. This is because the orthogonality technique adopted by DP-GCN cannot statistically decorrelate the sensitive and non-sensitive attributes.

7 Conclusion and Future Work

In this paper, we introduced encoder and decoder architectures to learn useful graph representations for downstream tasks while disentangling the learned representation from the sensitive attributes to provide privacy protection. We considered two problems based on whether or not the adversary has the graph structure as auxiliary information. Experiments verified the effectiveness of our method. Future works include relaxing the distribution assumption of the latent variable of the autoencoder for a more general formulation. Moreover, the selection of privacy candidates is also worth studying.

REFERENCES

C. Agarwal, H. Lakkaraju, and M. Zitnik. Towards a unified framework for fair and stable graph representation learning. In *Proc. Conf. Uncertainty in Artificial Intelligence*, Virtual, 2021.

S. Agarwal. Trade-offs between fairness and privacy in machine learning. In *Proc. Int. Joint Conf. Artificial Intelligence*, Virtual, August 2021.

F. P. Calmon and N. Fawaz. Privacy against statistical inference. In *Proc. Asilomar Conf. on Signals, Systems and Computers*, Monticello, IL, USA, October 2012.

Thomas A. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, NY, second edition, 2005.

E. Dai and S. Wang. Learning fair graph neural networks with limited and private sensitive attribute information. *IEEE Trans. Knowledge and Data Engineering*, 1(1):1–14, August 2022.

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

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. Conf. Theory of Cryptography*, New York, NY, USA, March 2006.

W. Fan, K. Liu, R. Xie, H. Liu, H. Xiong, and Y. Fu. Fair graph auto-encoder for unbiased graph representations with wasserstein distance. In *Proc. Int. Conf. Data Mining*, Auckland, New Zealand, December 2021.

N. Z. Gong and B. Liu. You are who you know and how you behave: Attribute inference attacks via users social friends and behaviors. In *Proc. USENIX Security Symposium*, Austin, TX, August 2016.

N. Z. Gong and B. Liu. Attribute inference attacks in online social networks. *ACM Trans. Privacy and Security*, 21(11):1–30, jan 2018.

T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto. Knowledge transfer for out-of-knowledge-base entities : A graph neural network approach. In *Proc. Int. Joint Conf. Artificial Intelligence*, Melbourne, Australia, August 2017.

W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proc. Int. Conf. Neural Information Processing Systems*, Long Beach, California, December 2017a.

W. L. Hamilton, R. Ying, and J. Leskovec:. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017b.

H. Hu, L. Cheng, J. P. Vap, and M. Borowczak. Learning privacy-preserving graph convolutional network with partially observed sensitive attributes. In *Proc. ACM Web Conf.*, Lyon, France, April 2022.

C. Huang, P. Kairouz, and L. Sankar. Generative adversarial privacy: A data-driven approach to information-theoretic privacy. In *Proc. Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, USA, USA, October 2018.

D. P. Kingma and M. Welling. Auto-Encoding variational bayes. In *Proc. Int. Conf. Learning Representations*, Banff, Canada, April 2014.

T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learning Representations*, Toulon, France, April 2017.

K. Li, G. Luo, Y. Ye, W. Li, S. Ji, and Z. Cai. Adversarial privacy-preserving graph embedding against inference attack. *IEEE J. Internet of Things*, 8(8):6904–6915, April 2021.

C. Louizos, K. Swersky, Y. J. Li, M. Welling, and R. Zemel. The variational fair autoencoder. In *Proc. Int. Conf. Learning Representations*, San Juan, Puerto Rico, May 2016.

A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. In *Proc. Int. Conf. Data Eng.*, Atlanta, GA, USA, April 2006.

M. McPherson, L. Smith-Lovin, and J. M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, August 2001.

D. Moyer, S. Y. Gao, R. Brekelmans, G. V. Steeg, and A. Galstyan. Invariant representations without adversarial training. In *Proc. Int. Conf. Neural Information Processing Systems*, Montreal, Canada, December 2018.

D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer. From t-closeness-like privacy to postrandomization via information theory. *IEEE Trans. Knowledge and Data Engineering*, 22(11):1623–1636, November 2010.

L. Sankar, S. R. Rajagopalan, and H. V. Poor. Utility-privacy tradeoffs in databases: An information-theoretic approach. *IEEE Trans. Information Theory*, 8(6):838–852, June 2013.

K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Proc. Conf. Neural Information Processing System*, Montréal, Canada, December 2015.

L. Sun, Y. Dou, C. Yang, K. Zhang, J. Wang, P. S. Yu, L. He, and B. Li. Adversarial attack and defense on graph data: A survey. *IEEE Trans. Knowledge and Data Engineering*, 1(1):1–20, 2022.

L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

L. Takac and M. Zabovsky. Data analysis in public social networks. In *Proc. Int. Conf. Innovation and New Trends in Information Technology*, Lomza, Poland, May 2012.

Y. Wang, Y. Zhao, Y. Dong, H. Chen, J. Li, and T. Derr. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In *Proc. Conf. Knowledge Discovery and Data Mining*, Washington, D.C., August 2022.

R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proc. Int. Conf. Knowledge Discovery and Data Mining*, London, United Kingdom, July 2018.

J. You, X. Ma, D. Y. Ding, M. Kochenderfer, and J. Leskovec. Handling missing data with graph representation learning. In *Proc. Int. Conf. Neural Information Processing Systems*, Red Hook, NY, USA, December 2020.

X. Zhang and M. Zitnik. GNNGUARD: Defending graph neural networks against adversarial attacks. In *Proc. Int. Conf. Neural Information Processing Systems*, Red Hook, NY, USA, December 2020.

J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L.Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 11:57–81, April 2020.

A Privacy Quantification

In this section, we show that our privacy penalty parameter $\Delta\boldsymbol{\nu}$ in Section 5.2 bounds the mutual information between $\mathbf{Z}$ and $\mathbf{s} \in \mathcal{S}$, and hence can limit the accuracy of detecting $\mathbf{s}_{\mathcal{P}}$ based on $\mathbf{Z}$ for an adversary for case (b). The derived results naturally apply to case (a) also.

A.1 Divergence Inequalities

We derive some inequalities for our analysis. The Kullback-Leibler (KL) divergence (also known as relative entropy) is a measure of how one probability distribution is different from a reference probability distribution (Cover & Thomas, 2005). The KL divergence between two distributions p and q over a sample space Ω is defined as

$$D_{\mathrm{KL}}(p \,\|\, q) = \int_{\Omega} p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}x.$$

Note $D_{\mathrm{KL}}(p \,\|\, q) = 0$ if and only if $p = q$ almost everywhere.

Let p' and q' be two distributions over Ω. From the property of joint convexity (which holds for any f-divergence) (Cover & Thomas, 2005), we have

$$D_{\mathrm{KL}}(\lambda p + (1-\lambda)p' \,\|\, \lambda q + (1-\lambda)q') \leq \lambda D_{\mathrm{KL}}(p \,\|\, q) + (1-\lambda) D_{\mathrm{KL}}(p' \,\|\, q'), \tag{7}$$

for any $0 \leq \lambda \leq 1$. Now letting $q' = p' = p$ in (7), we immediately obtain

$$\begin{aligned} D_{\mathrm{KL}}(p \,\|\, \lambda q + (1-\lambda)p) &\leq \lambda D_{\mathrm{KL}}(p \,\|\, q) + (1-\lambda) D_{\mathrm{KL}}(p \,\|\, p) \\ &= \lambda D_{\mathrm{KL}}(p \,\|\, q). \end{aligned} \tag{8}$$

By inductive reasoning, (8) generalizes to

$$D_{\mathrm{KL}}(p \,\|\, \lambda_1 q + \sum_{i=2}^{K} \lambda_i p_i) \leq \sum_{i=2}^{K} \lambda_i D_{\mathrm{KL}}(p \,\|\, q_i), \tag{9}$$

where $q_2, \ldots, q_K$ are distributions over Ω and $\sum_{i=1}^{K} \lambda_i = 1$ with $\lambda_i \geq 0$.

The mutual information (MI) is a measure of the statistical dependence between two random variables, which quantifies the amount of information in units of shannons bits obtained about one random variable by observing the other random variable (Cover & Thomas, 2005). Consider random variables x and y, with $p_{\mathrm{x,y}}(\mathrm{x}, \mathrm{y})$ being their joint distribution and $p_{\mathrm{x}}(\mathrm{x})$ and $p_{\mathrm{y}}(\mathrm{y})$ being the marginal distributions of x and y, respectively. Let $q_{\mathrm{x,y}}(\mathrm{x}, \mathrm{y}) = p_{\mathrm{x}}(\mathrm{x}) p_{\mathrm{y}}(\mathrm{y})$. The MI between x and y is written as

$$\begin{aligned} I(\mathrm{x}; \mathrm{y}) &= D_{\mathrm{KL}}(p_{\mathrm{x,y}} \,\|\, q_{\mathrm{x,y}}) \\ &= \mathbb{E}_{\mathrm{y} \sim p_{\mathrm{y}}(\mathrm{y})} \big[D_{\mathrm{KL}}(p_{\mathrm{x}|\mathrm{y}}(\cdot \mid \mathrm{y}) \,\|\, p_{\mathrm{x}}) \big]. \end{aligned}$$

A.2 Bounding Mutual Information

In what follows, we show that the MI between the graph embedding $\mathbf{Z}$ and the sensitive attribute $\mathbf{s}$ is bounded by our privacy penalty parameter $\Delta\boldsymbol{\nu}$. Suppose $\mathbf{s} \in \mathcal{S}$ with $|\mathcal{S}| < \infty$, where $\mathcal{S}$ is the set of privacy candidates of $\boldsymbol{s}$.

Recall $p(\mathbf{Z} \mid \mathbf{s}) = \prod_{i=1}^{N} p(\mathbf{z}_i \mid \mathrm{s}_i)$ with $p(\mathbf{z}_i \mid \mathrm{s}_i) = \mathcal{N}(\mathbf{z}_i \,|\, \boldsymbol{\nu}(\mathrm{s}_i), \mathbf{I}_M)$ for case (b). The marginal distribution of $\mathbf{Z}$ can be written as

$$p(\mathbf{Z}) = \sum_{\boldsymbol{s}' \in \mathcal{S}} p(\mathbf{s} = \boldsymbol{s}') p(\mathbf{Z} | \mathbf{s} = \boldsymbol{s}').$$

From the inequality (9), for $\boldsymbol{s}' \in \mathcal{S}$, we have

$$\begin{aligned} D_{\mathrm{KL}}(p(\mathbf{Z}|\mathbf{s} = \boldsymbol{s}') \,\|\, p(\mathbf{Z})) &\leq \sum_{\boldsymbol{s}'' \in \mathcal{S}} p(\mathbf{s} = \boldsymbol{s}'') D_{\mathrm{KL}}(p(\mathbf{Z}|\mathbf{s} = \boldsymbol{s}') \,\|\, p(\mathbf{Z}|\mathbf{s} = \boldsymbol{s}'')) \\ &= \frac{1}{2} \sum_{\boldsymbol{s}'' \in \mathcal{S}} p(\mathbf{s} = \boldsymbol{s}'') \sum_{i=1}^{N} \|\boldsymbol{\nu}(\boldsymbol{s}_i') - \boldsymbol{\nu}(\boldsymbol{s}_i'')\|^2 \\ &\leq \frac{1}{2} \sum_{\boldsymbol{s}'' \in \mathcal{S}} p(\mathbf{s} = \boldsymbol{s}'') (2\Delta\boldsymbol{\nu}) = \Delta\boldsymbol{\nu}. \end{aligned}$$

Subsequently, we have

$$\begin{aligned} I(\mathbf{Z}; \mathbf{s}) &= \mathbb{E}_{\mathbf{s} \sim p(\mathbf{s})} [D_{\mathrm{KL}}(p(\mathbf{Z}|\mathbf{s}) \,\|\, p(\mathbf{Z}))] \\ &= \sum_{\boldsymbol{s}' \in \mathcal{S}} p(\mathbf{s} = \boldsymbol{s}') D_{\mathrm{KL}}(p(\mathbf{Z}|\mathbf{s} = \boldsymbol{s}') \,\|\, p(\mathbf{Z})) \leq \Delta\boldsymbol{\nu}. \end{aligned}$$

It can be concluded now that the privacy penalty $\Delta\boldsymbol{\nu}$ serves as an upper-bound of the MI between $\mathbf{Z}$ and $\mathbf{s}$.

A.3 DETECTION ERROR

Now we quantify the adversary's loss of inferring $\mathbf{s}_{\mathcal{P}}$ based on graph embedding $\mathbf{Z}$. We note the Fano's inequality (Cover & Thomas, 2005):

$$H(\mathbf{s} \mid \mathbf{X}) \leq \mathbb{P}(e) \log(|\mathcal{S}| - 1),$$

where $\mathbb{P}(e) = \mathbb{P}(\widehat{\mathbf{s}}(\mathbf{Z}) \neq \mathbf{s})$ denotes the detection error of $\mathbf{s}$ based on $\mathbf{Z}$ and $H(\cdot \mid \cdot)$ denotes the conditional entropy function.

Let $H(\cdot)$ be the entropy function. It can be easily verified that $H(\mathbf{s}) \geq H(p_{\min}(\mathbf{s}))$ where $p_{\min}(\mathbf{s}) = \min_{\boldsymbol{s} \in \mathcal{S}} p(\mathbf{s} = \boldsymbol{s})$ and $H(p_{\min}(\mathbf{s}))$ is the binary entropy function w.r.t. $p_{\min}(\mathbf{s})$. Note $H(p_{\min}(\mathbf{s}))$ is a monotonically increasing function w.r.t. $p_{\min}(\mathbf{s})$.

From

$$I(\mathbf{X}; \mathbf{s}) = H(\mathbf{s}) - H(\mathbf{s} \mid \mathbf{X}),$$

we obtain

$$\mathbb{P}(e) \geq \frac{H(p_{\min}(\mathbf{s})) - I(\mathbf{X}; \mathbf{s})}{\log(|\mathcal{S}| - 1)} \geq \frac{H(p_{\min}(\mathbf{s})) - \Delta\boldsymbol{\nu}}{\log(|\mathcal{S}| - 1)}. \tag{10}$$

Inequality (10) implies that the detection error increases as $\Delta\boldsymbol{\nu}$ decreases, leading to stronger privacy. However, we need to bound the prior of the sensitive attribute $p_{\min}(\mathbf{s})$ to ensure this. This result justifies our sanitization strategy.