Neurocomputing 561 (2023) 126841

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Learning fair models without sensitive attributes: A generative approach

Huaisheng Zhu [a], Enyan Dai [a], Hui Liu [b], Suhang Wang [a,*]

[a] College of Information Science and Technology, The Pennsylvania State University, University Park, 16802, PA, United States
[b] Department of Computer Science and Engineering, Michigan State University, East Lansing, 48824, MI, United States

ARTICLE INFO

Communicated by L. Oneto

Keywords:
Fairness
Generative model

ABSTRACT

Most existing fair classifiers rely on sensitive attributes to achieve fairness. However, for many scenarios, we cannot obtain sensitive attributes due to privacy and legal issues. The lack of sensitive attributes challenges many existing fair classifiers. Though we lack sensitive attributes, for many applications, there usually exists features/information of various formats that are relevant to sensitive attributes. For example, a person's purchase history can reflect his/her race, which would help for learning fair classifiers on race. However, the work on exploring relevant features for learning fair models without sensitive attributes is rather limited. Therefore, in this paper, we study a novel problem of learning fair models without sensitive attributes by exploring relevant features. We propose a probabilistic generative framework to effectively estimate the sensitive attribute from the training data with relevant features in various formats and utilize the estimated sensitive attribute information to learn fair models. Experimental results on real-world datasets show the effectiveness of our framework in terms of both accuracy and fairness. Our source code is available at: https://github.com/huaishengzhu/FairWS.

1. Introduction

Over the past few years, machine learning models have shown great success in a wide spectrum of applications, such as credit scoring [1], crime prediction [2], and salary prediction [3]. However, there is a growing concern about societal bias in training data on demographic or sensitive attributes such as age, gender and race [4,5]. In particular, machine learning models trained on biased data can inherit the bias or even reinforce it. For example, a strong unfairness is found in the software COMPAS, which is used to predict the risk of a criminal to recommitting another crime [2]. It is found that COMPAS is more likely to assign a higher risk score to criminals of color even when they do not recommit another crime. Thus, bias issues in a machine learning model could cause severe fairness problems, which raises concerns about their real-world applications, especially in high-stake scenarios such as credit scoring and crime prediction.

Therefore, extensive studies have been conducted to mitigate the bias issues of machine learning models [6,7], which can be generally categorized into three categories, i.e., pre-processing, in-processing, and post-processing. Pre-processing approaches process the training data to remove discrimination. For example, they can reduce the bias in the data by revising the attributes [8], generating non-discriminatory labeled data [9], and learning fair representations [10]. In-processing approaches will modify the training process of the state-of-the-art model. Typically, in-processing methods incorporate fairness constraint /regularizer into the objective function of the model, which can mitigate the bias of the models' prediction results [7,11]. As for post-processing algorithms, they directly change the predictions from the trained model to meet the requirement of fairness [5,12].

Despite their effectiveness, the aforementioned approaches generally require the protected/sensitive attributes of each data sample to preprocess the data, regularize the model or post-process the predictions to achieve fairness. However, for many real-world applications, obtaining sensitive attributes is difficult due to privacy and legal issues [13,14]. For example, Consumer Financial Protection Bureau (CFBP) requires that creditors may not collect information about an applicant's race, color, religion and other sensitive information [14]. Another scenario is the dataset collector did not realize the potential bias issue when the dataset was built. Hence, the protected/sensitive attributes which would be useful to mitigate the bias issue of machine learning models are not collected. The lack of sensitive attributes challenges most existing fairness-aware machine learning models as they rely on sensitive attributes to achieve fairness. There are only very few initial works on training fair classifiers without sensitive attributes [14–16]. For example, Yan et al. [16] introduces a clustering algorithm to

* Corresponding author.
E-mail address: szw494@psu.edu (S. Wang).

https://doi.org/10.1016/j.neucom.2023.126841
Received 20 February 2023; Received in revised form 5 July 2023; Accepted 23 September 2023
Available online 27 September 2023

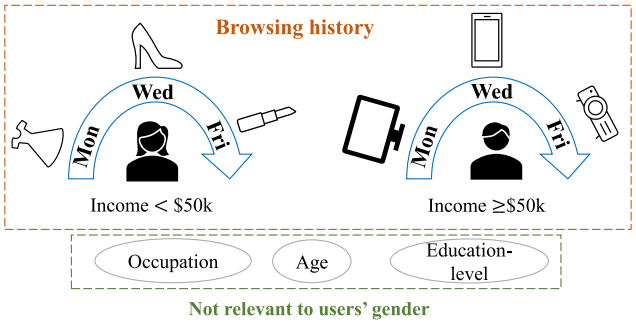


Fig. 1. Illustration of irrelevant features and relevant features (browsing history) w.r.t the sensitive attribute (gender) together with their labels (income).

obtain pseudo groups to replace the real protected groups. However, it cannot guarantee that the groups they find are relevant to targeted subgroups to be protected. To resolve this problem, Zhao et al. [15] assume that non-sensitive features that are highly correlated with sensitive attributes exist in the dataset and treat these non-sensitive features as pseudo sensitive attributes. Though effective, it is a strong assumption that these relevant features are highly correlated with sensitive attributes. Therefore, how to address fairness issues without knowing the sensitive attribute of each data sample is still an open problem to be addressed.

Though we lack sensitive attributes, for many real-world applications, there usually exists features/information of various formats that are not only relevant to class labels but also have a high dependency on sensitive attributes. For example, a person's purchase or dining history can reflect the person's cultural background [17]. A user's linguistic style of reviews or browsing history can indicate the user's gender [18, 19]. We call such feature/information relevant to sensitive attributes as *relevant features* and the remaining features as *irrelevant features*. Note that those *irrelevant features* might have a very weak dependency with the sensitive attributes, but would not be useful to infer sensitive attributes. Since those sensitive attributes can be in various formats, e.g., texts and graphs, they cannot be directly used as pseudo-sensitive attributes. However, it is possible to estimate sensitive attributes from those relevant features, which would be useful to train a fair classifier. As the example shown in Fig. 1, women prefer to browse dresses, high-heeled shoes and makeup; while men tend to search electronic devices on the website. In this case, we can easily infer users' gender which is a sensitive attribute from their browsing history. In addition to browsing history, there are other attributes such as occupation, age and education-level, which are irrelevant to the user's gender but can provide useful information for the income prediction task.

One way to alleviate bias issues is to only use those irrelevant features for prediction. However, it has several issues: (i) the labels collected might already contain bias. Such bias cannot be reduced without using sensitive attribute information to guide model learning; (ii) the relevant features are also useful for the prediction tasks. For example, items in browsing history also imply the income (the label) of the users by their price. Therefore, it is necessary to combine both browsing history and irrelevant features to train an accurate classifier. Therefore, though no sensitive attributes are provided for debiasing, it is promising to utilize relevant features to estimate sensitive attributes and adapt them to regularize the classifier for fair predictions. Meanwhile, the classifier utilizes both relevant and irrelevant features for better prediction accuracy. However, the work on this is rather limited.

Therefore, in this paper, we study a novel problem of learning fair classifiers without sensitive attributes by estimating sensitive information from training data. There are several challenges: (i) how can we effectively estimate sensitive information given that the sensitive attributes have a dependency on both relevant features and labels? and (ii) how to incorporate the estimated sensitive information to learn fair classifiers? To fill this gap, we propose a novel framework Fair Models Without Sensitive Attributes (FairWS). It adopts a probabilistic graphical model to capture the dependency between sensitive attributes, relevant features, irrelevant features and labels. Then, a Variational Autoencoder is used to model these relationships in the probabilistic graphical model, which paves a way to effectively estimate the sensitive attributes. To ensure the fairness of the given predictions, FairWS designs a fairness regularization term based on the estimated sensitive information. The main contributions are as follows:

- We study a novel problem of learning fair classifiers without sensitive attributes by estimating sensitive information from the training data;
- We propose a new framework FairWS, which is flexible to estimate sensitive information from relevant features in various formats such as texts and graphs and utilize the inferred sensitive information to regularize existing classifiers to achieve fairness;
- We conduct experiments on real-world datasets with relevant features in various formats to show the effectiveness of FairWS for fair and high accuracy classification.

2. Related work

In this section, we review related works, including fairness in machine learning and deep generative models.

2.1. Fairness in machine learning

Recent studies [5,8,11] show that machine learning models can inherit societal bias from historical data. Thus, learning fair machine learning models has attracted increasing attention and many efforts have been taken [14], which can be generally split into three categories: (i) individual fairness [11,20–22], which trains the model to provide similar individuals with similar predictions; (ii) group fairness [5,11,23], which requires the model to give equal prediction to groups with various protected sensitive attributes; (iii) Max-Min fairness [14,24–26], which aims to maximize the minimum expected utility across protected groups. We focus on group fairness in this work.

Based on the stage of achieving fairness, existing fair machine learning methods can be split into three categories, i.e, pre-processing, in-processing, and post-processing [2]. Pre-processing methods [8–10] reduce the historical discrimination in the dataset by modifying the training data. For example, Feldman et al. [8] introduce an approach to revise the attributes of training data and Xu et al. [9] propose to generate non-discriminatory data. Locatello et al. [10] obtain fair representation for unbiased prediction. In-processing methods [7,11] revise the training of fair machine learning models by designing fairness constraints or objective functions to train fair models. Post-processing methods [5,12] modify the prediction results from the training models to achieve fairness.

Despite their effectiveness in mitigating bias issues, the aforementioned methods require sensitive attributes of each data sample to achieve fairness, while for many scenarios, obtaining sensitive attributes is difficult due to various issues, which challenge existing fair models. Specifically, developing a fair model without sensitive attributes is crucial in real-world scenarios where obtaining sensitive attributes is important due to privacy and legal concerns [13, 14]. Collecting sensitive attributes can raise ethical issues and violate privacy regulations in many domains, including health care [27], credit scoring [28] and criminal justice [29]. There is limited work on learning fair models without sensitive attributes [14–16]. Lahoti et al. [14] propose an optimization approach that leverages the notion of computationally-identifiable errors and improves the utility for worst-off protected groups. Yan et al. [16] conducts clustering to obtain pseudo groups to substitute the real protected groups. However, the groups found by it may be irrelevant to the sensitive attributes we want

the model to be fair with. For example, we might aim to make the model fair with Race but clustering gives pseudo groups for Gender. Thus, another trend of work assumes some prior knowledge about sensitive information so that their model can be fair with targeted sensitive attributes. For instance, Zhao et al. [15] assume that there are some features strongly correlated with the sensitive attributes and directly utilize these features as pseudo sensitive attributes. However, such strongly correlated features that can be treated as pseudo-sensitive attributes are not always available in real-world applications. Moreover, Grari et al. [30] introduce a generative model to learn a fair model (SRCVAE) without the need for sensitive attributes, which is the most similar work to ours. The proposed approach, SRCVAE, aims to extract latent sensitive information from attributes that are influenced by sensitive attributes. However, it is important to note that attributes resulting from sensitive attributes may not exclusively contain sensitive information, as they can also incorporate non-sensitive information. Therefore, one limitation of their approach is the inability to effectively disentangle sensitive information from non-sensitive information in features that are influenced by sensitive attributes.

Our proposed FairWS is inherently different from the aforementioned approaches: (i) Instead of directly using the relevant features to obtain pseudogroups, we estimate sensitive information from relevant features to train fair and accurate classifiers. And little prior knowledge is required in FairWS to infer the sensitive information; and (ii) FairWS is flexible to learn sensitive information from relevant features of various formats, like texts and graphs. (iii) FairWS introduces a loss based on mutual information to enable the disentanglement of sensitive information from non-sensitive information in features that are influenced by sensitive attributes so that it can effectively extract relevant sensitive information from relevant features. The inferred sensitive information can be utilized to regularize existing fair models. Also, FairWS can infer sensitive information from noisy relevant features in our experiments .

2.2. Deep generative model

Generative models aim to capture the underlying data distribution. Due to their superior performance, deep generative models like VAE [31] and GAN [32] have attracted increasing attention. Hu et al. [33] provides a unified view of various deep generative models. Furthermore, there are many efforts taken to generate real data based on GANs and VAEs [34–38]. For example, GANs are utilized to generate realistic images [38]. Controlled generation of text based on VAE has been explored [34,37]. Recently, there are some efforts of using VAEs to resolve the fairness problem [39–42]. Firstly, Variational Fair Autoencoder [40] is proposed to build a probabilistic graphical model to model data with sensitive attributes. Then, a fair latent representation is obtained by removing sensitive information. Creager et al. [39] propose to learn the latent representation of VAEs by disentangling it into two parts based on whether they are relevant to sensitive attributes. And representations irrelevant to sensitive attributes can be used as fair representation to learn fair models. Amini et al. [41] apply fair VAEs to learn latent fair representations in facial detection systems. However, works about exploring the ability of VAEs to mitigate fairness problems without sensitive attributes are rather limited. Moreover, the proposed FairWS is a general framework to infer sensitive information from dependency on relevant features with different formats and sensitive attributes.

3. Problem definition and notations

For many real-world applications, sensitive attributes of data samples are unavailable due to various issues such as difficulty in data collection, security or privacy issues. The lack of sensitive attributes challenges existing fairness-aware machine learning models that require sensitive attributes of data samples to achieve fairness. Though sensitive attributes for many real-world applications are unavailable, we observe that there is usually information highly relevant to sensitive attributes. For example, a person's purchase or dining history can reflect the person's gender or cultural background, which would help learn fair classifiers on gender or cultural background. We call such information *relevant features* and the remaining features that is not related to the sensitive attribute to be protected as *irrelevant features*. Note that the relevant features can be in various formats such as sequences (purchase history) and texts (reviews). Thus, the relevant features cannot be directly utilized as pseudo-sensitive attributes to regularize the model [15]. In this paper, we aim to utilize the relevant feature to estimate sensitive attributes and train a fair and accurate classifier using both relevant and irrelevant features with estimated sensitive attributes.

Specifically, we use $D = \{\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i\}_{i=1}^N$ to denote the training set with N data samples, where $(\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ is the ith sample. $\mathbf{x}_i^z$ is a feature vector that is not relevant or is very weakly related to the protected sensitive attribute $\mathbf{S}$ such as gender, $\mathbf{x}_i^r$ is relevant feature vector that are related with $\mathbf{S}$, and $\mathbf{y}_i$ is the class label. $\mathbf{x}_i^r$ can be in various formats such as purchase history and review texts. In this paper, we focus on fairness concerning a single sensitive attribute $\mathbf{S}$. We leave the extension to multi-sensitive attributes as future work. Note that we do not know the values of the sensitive attribute of each data sample. The problem is formally defined as:

Given the training set $D_l = \{\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i\}_{i=1}^N$ with $\mathbf{x}_i^r$ being relevant feature w.r.t protected sensitive attribute $\mathbf{S}$ and $\mathbf{x}_i^z$ being irrelevant features, we aim to learn a fair and accurate classifier $f(\mathbf{x}^z, \mathbf{x}^r) \rightarrow \hat{\mathbf{y}}$, where f denotes the function to learn and $\hat{\mathbf{y}}$ represents the prediction from the classifier. And the set of predictions on test set should simultaneously maintain high accuracy and meet the fairness criteria w.r.t to the sensitive attribute $\mathbf{S}$.

4. Proposed framework

In this section, we introduce the details of the proposed FairWS for learning fair models without sensitive attributes. Without sensitive attributes to achieve fairness, our basic idea is to estimate sensitive attributes by exploring relevant features and adopting the estimated sensitive attributes for learning fair classifiers. However, how to effectively estimate sensitive information given that the sensitive attributes have a dependency on both relevant features and labels remains a question. To fully capture the dependency for accurately estimating sensitive attributes, we assume that each observed data $(\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ is sampled from a probabilistic generative process which involves the latent sensitive information $\mathbf{a}_i$ and latent data representation $\mathbf{z}_i$. Thus, FairWS models the probabilistic generative process to estimate sensitive information and utilizes the estimated sensitive information to learn fair classifiers. Another important issue is about how to incorporate the estimated sensitive information to learn fair classifiers. To resolve this problem, we introduce a regularization term to train a fair classifier with the generated latent representation of sensitive attributes. We will first introduce the probabilistic generative model for sensitive attribute estimation followed by fair classifier learning.

4.1. Sensitive attributes estimation

Though the relevant features $\mathbf{x}_i^r$ are related to the sensitive attribute $\mathbf{s}_i$ of the ith data sample, they cannot be simply treated as pseudo-sensitive attributes as $\mathbf{x}_i^r$ could be in various formats and might be noisy. Meanwhile, both relevant features and labels have a dependency on sensitive attributes, which can be used to estimate the latent sensitive attributes for fair classifiers. To handle this, we use a probabilistic graphical model that models the dependency relations to obtain latent sensitive attributes. The advantages are: (i) probabilistic graphical model can capture the complex relationships among sensitive attribute $\mathbf{s}_i$, label $\mathbf{y}_i$, relevant feature $\mathbf{x}_i^r$ and irrelevant feature $\mathbf{x}_i^z$, which can help better estimate the sensitive attribute information; and (ii) estimated

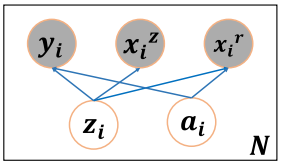


Fig. 2. Probabilistic Graphical Model of FairWS.

sensitive attribute information can be easily adopted to learn fair classifiers.

Specifically, we assume that each data sample $(\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ is sampled from a generative process as shown in Fig. 2, where $\mathbf{z}_i$ is the intrinsic latent representation irrelevant to sensitive attributes $\mathbf{s}_i$, and $\mathbf{a}_i$ is the latent representation of the sensitive attribute $\mathbf{s}_i$. $\mathbf{x}_i^r$ is dependent on sensitive attribute's latent representation $\mathbf{a}_i$ as $\mathbf{x}_i^r$ is relevant with $\mathbf{s}_i$. As the collected labels generally contain a bias towards the sensitive attribute, we assume that $\mathbf{y}_i$ is also dependent on $\mathbf{a}_i$. Since $\mathbf{z}_i$ contains the intrinsic characteristic of data sample i, the class label $\mathbf{y}_i$ is dependent on $\mathbf{z}_i$. $\mathbf{x}_i^r$ is also dependent on the $\mathbf{z}_i$ because $\mathbf{x}_i^r$ also contains some information that is irrelevant to sensitive attributes. It is worth noting that $\mathbf{x}_i^z$ is independent with $\mathbf{s}_i$ or has a very weak (neglectable) relationship with $\mathbf{s}_i$ because those features that are highly relevant to $\mathbf{s}_i$ are already included in $\mathbf{x}_i^r$. Hence, there is no dependency between $\mathbf{x}_i^z$ and $\mathbf{a}_i$. We disentangle $\mathbf{a}_i$ and $\mathbf{z}_i$ so that $\mathbf{a}_i$ and $\mathbf{z}_i$ can extract sensitive attribute information and non-sensitive information from $\mathbf{x}_i^r$, respectively.

According to the probabilistic graphical model in Fig. 2, $\mathbf{x}_i^z$ is independent to the latent representation of sensitive attributes $\mathbf{a}_i$, and $\mathbf{x}_i^r$ is highly correlated with $\mathbf{z}_i$. Then, the joint distribution $p(\mathbf{a}_i, \mathbf{y}_i, \mathbf{z}_i, \mathbf{x}_i^r, \mathbf{x}_i^z)$ can be written as:

$$\begin{aligned} & p(\mathbf{a}_i, \mathbf{y}_i, \mathbf{z}_i, \mathbf{x}_i^r, \mathbf{x}_i^z) \\ = \; & p(\mathbf{a}_i)p(\mathbf{z}_i)p(\mathbf{x}_i^r \mid \mathbf{a}_i, \mathbf{z}_i)p(\mathbf{x}_i^z \mid \mathbf{z}_i)p(\mathbf{y}_i \mid \mathbf{z}_i, \mathbf{a}_i), \end{aligned} \tag{1}$$

where $p(\mathbf{a}_i)$ and $p(\mathbf{z}_i)$ are the prior distributions, which are usually implemented as standard Gaussian distributions. Our goal is to maximize the likelihood of the joint distribution of observed variables, i.e., $p_\theta(\mathbf{x}_i^r, \mathbf{x}_i^z, \mathbf{y}_i)$. However, directly maximizing it is difficult as it contains latent variables $\mathbf{z}_i$ and $\mathbf{a}_i$. Following VAE [31], we maximize the variational lower bound of this likelihood as:

$$\begin{aligned} \log p_\theta(\mathbf{x}_i^r, \mathbf{x}_i^z, \mathbf{y}_i) \geq \; & \mathbb{E}_{q_\phi(\mathbf{a}_i, \mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)}[p_\theta(\mathbf{x}_i^r, \mathbf{x}_i^z, \mathbf{y}_i \mid \mathbf{z}_i, \mathbf{a}_i)] \\ & - D_{KL}(q_\phi(\mathbf{a}_i, \mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i) \parallel p(\mathbf{z}_i, \mathbf{a}_i)), \end{aligned} \tag{2}$$

where $\mathbb{E}$ denotes the expectation, $q_\phi(\mathbf{a}_i, \mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ is an auxiliary distribution to approximate $p_\theta(\mathbf{a}_i, \mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$, and $D_{KL}(\cdot, \cdot)$ denotes the Kullback–Leibler divergence. ϕ and θ are learnable parameters for neural networks of p and q, respectively.

As our goal is to disentangle $\mathbf{a}_i$ and $\mathbf{z}_i$, for $q_\phi(\mathbf{a}_i, \mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$, we also assume that $\mathbf{a}_i$ and $\mathbf{z}_i$ are independent given $\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i$, i.e.,

$$q_\phi(\mathbf{a}_i, \mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i) = q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^z, \mathbf{y}_i)q_\phi(\mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i), \tag{3}$$

where $q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^z, \mathbf{y}_i)$ and $q_\phi(\mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ can be treated as the encoders to learn the latent representation $\mathbf{a}_i$ and $\mathbf{z}_i$ from $(\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$.

Based on Fig. 2, $p_\theta(\mathbf{x}_i^r, \mathbf{x}_i^z, \mathbf{y}_i | \mathbf{z}_i, \mathbf{a}_i)$ can be further written as:

$$p_\theta(\mathbf{x}_i^r, \mathbf{x}_i^z, \mathbf{y}_i | \mathbf{z}_i, \mathbf{a}_i) = p_\theta(\mathbf{x}_i^r | \mathbf{a}_i, \mathbf{z}_i)p_\theta(\mathbf{x}_i^z | \mathbf{z}_i)p_\theta(\mathbf{y}_i | \mathbf{z}_i, \mathbf{a}_i)$$

where $p_\theta(\mathbf{x}_i^r \mid \mathbf{a}_i, \mathbf{z}_i)$, $p_\theta(\mathbf{y}_i \mid \mathbf{z}_i, \mathbf{a}_i)$ and $p_\theta(\mathbf{x}_i^z \mid \mathbf{z}_i)$ are the decoders to generate $\mathbf{x}_i^r$ $\mathbf{y}_i$ and $\mathbf{x}_i^z$, respectively. The details of encoders and decoders will be discussed in Section 4.3. Assume that the prior distribution $p(\mathbf{z}_i, \mathbf{a}_i)$ can be factorized as $p(\mathbf{z}_i)p(\mathbf{a}_i)$ with both $p(\mathbf{z}_i)$ and $p(\mathbf{a}_i)$ follow the normal distribution, then we can rewrite the KL divergence as:

$$\begin{aligned} & D_{KL}(q_\phi(\mathbf{z}_i, \mathbf{a}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i) \parallel p(\mathbf{z}_i, \mathbf{a}_i)) \\ = \; & D_{KL}(q_\phi(\mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i) \| p(\mathbf{z}_i)) + D_{KL}(q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^r, \mathbf{y}_i) \| p(\mathbf{a}_i)). \end{aligned} \tag{4}$$

where $D_{KL}(q_\phi(\mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i) \parallel p(\mathbf{z}_i))$ and $D_{KL}(q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^r, \mathbf{y}_i) \parallel p(\mathbf{a}_i))$ are two KL divergence terms to regularize $q_\phi(\mathbf{z}_i | \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ and $q_\phi(\mathbf{a}_i | \mathbf{x}_i^r, \mathbf{y}_i)$, respectively. To provide flexibility of our model, following [43], we add a weight hyperparameter β to control the influence of $D_{KL}(q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^r, \mathbf{y}_i) \parallel p(\mathbf{a}_i))$. Then the variational lower bound can be written as:

$$\begin{aligned} \mathcal{L}_{ELBO}^i = \; & \mathbb{E}_{q_\phi(\mathbf{a}_i, \mathbf{z}_i | \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)}[p_\theta(\mathbf{x}_i^r, \mathbf{x}_i^z, \mathbf{y}_i \mid \mathbf{z}_i, \mathbf{a}_i)] \\ & - D_{KL}(q_\phi(\mathbf{z}_i \mid \mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i) \parallel p(\mathbf{z}_i)) \\ & - \beta D_{KL}(q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^r, \mathbf{y}_i) \parallel p(\mathbf{a}_i)). \end{aligned} \tag{5}$$

Since both $\mathbf{z}_i$ and $\mathbf{a}_i$ is dependent on the relevant features $\mathbf{x}_i^r$, to make sure that $\mathbf{z}_i$ and $\mathbf{a}_i$ are disentangled, i.e., $\mathbf{z}_i$ captures the class label related information from $\mathbf{x}_i^r$ while $\mathbf{z}_i$ captures sensitive attribute related information, we add a regularizer to minimize the mutual information between $\mathbf{z}_i$ and $\mathbf{a}_i$, i.e.:

$$\min_{\theta, \phi} \; I(\mathbf{A}; \mathbf{Z}) = H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{Z}) \tag{6}$$

where $\mathbf{A}$ is a matrix with the ith row as $\mathbf{a}_i$ and $\mathbf{Z}$ is a matrix with the ith row as $\mathbf{z}_i$. $H(\cdot)$ denotes the entropy function and $I(\mathbf{A}, \mathbf{Z})$ measures dependencies between $\mathbf{A}$ and $\mathbf{Z}$. However, the mutual information $I(\mathbf{A}; \mathbf{Z})$ is difficult to calculate directly. We follow [44] to efficiently estimate the mutual information. The basic idea is to train a neural network Dis to distinguish between sample pairs from the joint distribution $p(\mathbf{a}_i, \mathbf{z}_i)$ and those from $p(\mathbf{a}_i)p(\mathbf{z}_i)$. Note that we do not need to do prior assumption on $p(\mathbf{a}_i, \mathbf{z}_i)$ and the mutual information can be approximated as:

$$\begin{aligned} I(\mathbf{A}; \mathbf{Z}) \approx \; & \mathbb{E}_{p(\mathbf{a}_i, \mathbf{z}_i)}[Dis(\mathbf{a}_i, \mathbf{z}_i)] - \log \mathbb{E}_{p(\mathbf{a}_i)p(\mathbf{z}_i)}[e^{Dis(\mathbf{a}_i, \tilde{\mathbf{z}}_i)}] \\ = \; & \mathcal{L}_{MI}, \end{aligned} \tag{7}$$

where $Dis(\mathbf{a}_i, \mathbf{z}_i)$ is a binary discriminator judging if $(\mathbf{a}_i, \mathbf{z}_i)$ is from $p(\mathbf{a}_i, \mathbf{z}_i)$ or $p(\mathbf{a}_i)p(\mathbf{z}_i)$. In practice, in each batch to train Dis with batch size M, we sample a set of $\{(\mathbf{a}_i, \mathbf{z}_i)\}_{i=1}^M$ from the combination of representation matrices $\mathbf{A}$ and $\mathbf{Z}$ to estimate the first term of Eq. (7). Then, we randomly shuffle the rows of Z to obtain the corrupted representation matrix $\tilde{\mathbf{Z}}$, $\{(\mathbf{a}_i, \tilde{\mathbf{z}}_i)\}_{i=1}^M$ is sampled from the combination of representation matrices $\mathbf{A}$ and $\tilde{\mathbf{Z}}$ for estimating the first term of Eq. (7).

With the ELBO in Eq. (5) and the mutual information regularizer in Eq. (7), the final objective function of our sensitive attributes estimation module is:

$$\min_{\theta, \phi} \frac{1}{N} \sum_{i=1}^{N} -\mathcal{L}_{ELBO}^i + \mathcal{L}_{MI}, \tag{8}$$

Once the model is trained, we can estimate each data sample i's latent representation of sensitive attributes by sampling from $q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^z, \mathbf{y}_i)$. An illustration of the sensitive attribute estimation framework is shown in the right part of Fig. 3, where each term can be implemented as a neural network. To facilitate efficient large-scale training, we adopt the reparameterization trick [31].

4.2. Fairness regularization

As shown in Fig. 3, we can obtain ith data sample's latent representation of sensitive attributes, $\mathbf{a}_i$, from the sensitive attributes estimation module. Then, we can use sampled $\mathbf{a}_i$ to regularize a base classifier to learn a fair classifier. One way to train a fair classifier is to only use $\mathbf{x}_i^z$. However, since both $\mathbf{x}_i^r$ and $\mathbf{x}_i^z$ contain useful information for predicting the label of the ith data sample, using $\mathbf{x}_i^z$ only will lose much useful information, resulting in poor classification performance. Thus, we use both $\mathbf{x}_i^r$ and $\mathbf{x}_i^z$ to predict the label distribution of the ith data sample as:

$$\hat{\mathbf{y}}_i = g_{w_1}(\mathbf{x}_i^z \oplus f_{w_2}(\mathbf{x}_i^r)) \tag{9}$$

where $\oplus$ is the concatenation operator of two vectors and $g_{w_1}(\cdot)$ is multi-layered perceptrons (MLPs) to predict the label. As $\mathbf{x}_i^r$ can be in various formats such as texts and graphs, $f_{w_2}(\cdot)$ is utilized to transform $\mathbf{x}_i^r$ to a vector. For example, $f_{w_2}(\cdot)$ is the Convolutional Neural Network for text reviews and Graph Convolutional Network for graph data. w_1 and w_2 are both trainable parameters of neural networks. Note that

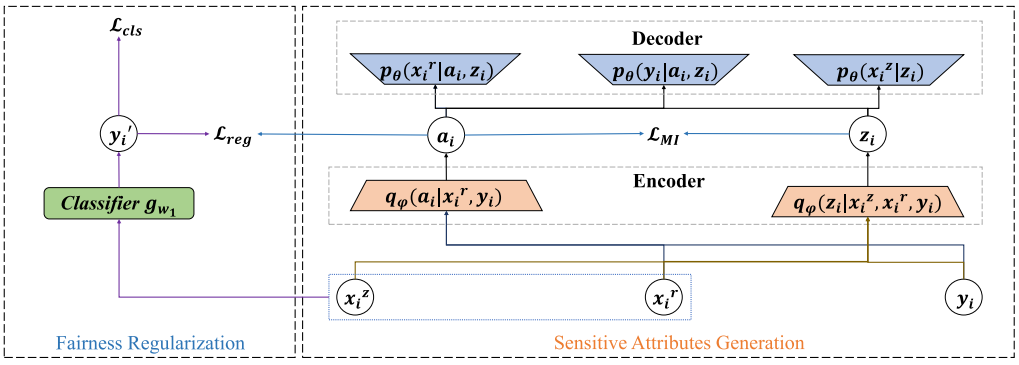


Fig. 3. An illustration of the proposed FairWS.

$f_{w_2}(\mathbf{x}_i^r) = \mathbf{x}_i^r$ if $\mathbf{x}_i^r$ is a relevant feature vector of the sample i. The cross-entropy loss for training the classifier g_{w_1} can be written as:

$$\min_{w_1,w_2} \mathcal{L}_{clf} = -\sum_{i=1}^{N}\sum_{j=1}^{m} \mathbf{y}_{ij} \log \hat{\mathbf{y}}_{ij}, \tag{10}$$

where $\mathbf{y}_i$ is one-hot encoding of the groundtruth label of $\mathbf{x}_i$ and m is the number of class. $\hat{\mathbf{y}}_{ij}$ denotes the predicted probability of i-the data sampled being class j.

Once well trained, the above classifier can give accurate predictions. However, $\mathbf{x}_i^r$ contains sensitive attribute information and the provided labels can also be biased, resulting in discriminatory predictions. To give fair predictions, we can utilize the estimated sensitive attributes to regularize the model. One way to make the prediction fair is to reduce the correlation between the prediction and the sensitive attribute $\mathbf{s}_i$. Since $\mathbf{a}_i$ is the latent representation of $\mathbf{s}_i$, the regularization form can be the correlation between predicted label vectors and latent sensitive attributes vectors as:

$$\mathcal{L}_{\text{reg}} = \sum_{k=1}^{d}\sum_{j=1}^{m} \left| \sum_{i=1}^{N} (\hat{\mathbf{y}}_{ij} - \bar{\mathbf{y}}_j)(\mathbf{a}_{ik} - \bar{\mathbf{a}}_k) \right|, \tag{11}$$

where N is the number of samples and m is the number of classes. d is the dimension of $\mathbf{a}_i$ and $\mathbf{a}_{ik}$ is the kth element of $\mathbf{a}_i$. $\bar{\mathbf{a}}_k = \frac{1}{N}\sum_{i=1}^{N} \mathbf{a}_{ik}$. $\hat{\mathbf{y}}_{ij}$ denotes the predicted probability of class j for sample i. And $\bar{\mathbf{y}}_j' = \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_{ij}$.

The final objective function of FairWS is given as:

$$\min_{w_1,w_2} \mathcal{L}_{clf} + \lambda \mathcal{L}_{reg}, \tag{12}$$

where λ is a scalar controlling the trade-off between the accuracy and fairness, and w_1 and w_2 are learnable parameters for transformation function and classifier in Eq. (9).

4.3. Deep learning framework of FairWS

With the generative model for sensitive attributes given above, we will introduce the details of modeling encoders $q_\phi(\mathbf{z}_i|\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ and $q_\phi(\mathbf{a}_i|\mathbf{x}_i^r, \mathbf{y}_i)$ together with decoders $p_\theta(\mathbf{x}_i^r|\mathbf{a}_i, \mathbf{z}_i)$, $p_\theta(\mathbf{x}_i^z|\mathbf{z}_i)$ and $p_\theta(\mathbf{y}_i|\mathbf{z}_i, \mathbf{a}_i)$ separately. In real-world applications, decoders and encoders can be very complex distributions for images and text data. In this paper, we adopt the reparameterization trick [31] and neural networks to model encoders and decoders, which can approximate complex distributions under mild conditions.

Firstly, we assume the encoders $q_\phi(\mathbf{z}_i|\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i)$ and $q_\phi(\mathbf{a}_i|\mathbf{x}_i^r, \mathbf{y}_i)$ both follow the Gaussian Distribution where mean and variance are the output of the neural network. It can be defined as:

$$\begin{aligned} q_\phi(\mathbf{z}_i|\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i) &= N(\mathbf{z}_i; \mu_{\mathbf{z}_i}, \sigma^2_{\mathbf{z}_i}\mathbf{I}) \quad \mu_{\mathbf{z}_i}, \sigma_{\mathbf{z}_i} = E_z(\mathbf{x}_i^z, \mathbf{x}_i^r, \mathbf{y}_i), \\ q_\phi(\mathbf{a}_i|\mathbf{x}_i^r, \mathbf{y}_i) &= N(\mathbf{a}_i; \mu_{\mathbf{a}_i}, \sigma^2_{\mathbf{a}_i}\mathbf{I}) \quad \mu_{\mathbf{a}_i}, \sigma_{\mathbf{a}_i} = E_a(\mathbf{x}_i^r, \mathbf{y}_i), \end{aligned} \tag{13}$$

Algorithm 1 Training Algorithm of FairWS.

Input: $\mathcal{D} = \{x_i^z, x_i^r, y_i\}_{i=1}^N$, λ and β.
Output: a fair classifier with f_{w_1} and g_{w_1}

```
Initialize parameters of E_z, E_a, D_{x^r}, D_{x^z} and D_y .
repeat
    Obtain labeled training samples {x_i^z, x_i^r, y_i}_{i=1}^N from D
    Optimized the encoder and decoder parameters E_z, E_a, D_{x^r}, D_{x^z}
    by Eq. (8).
until convergence
Infer the latent sensitive attributes A = {a_i}_{i=1}^N based on encoders
and decoders
Initialize parameters of f_{w_1} and g_{w_1}
repeat
    Get all labeled samples {x_i^z, x_i^r, y_i}_{i=1}^N from D and inferred
    sensitive latent representation A
    Optimize f_{w_1} and g_{w_1} by the loss from Eq. (12)
until convergence
return f_{w_1} and g_{w_1}
```

where $\mathbf{I}$ is the identity matrix, $E_z(\cdot)$ and $E_a(\cdot)$ are the neural networks. $E_z(\cdot)$ takes $\mathbf{x}_i^z, \mathbf{x}_i^r$ and $\mathbf{y}_i$ as input and outputs the mean $\mu_{\mathbf{z}_i}$ and standard deviation $\sigma_{\mathbf{z}_i}$. Similarly, $E_a(*)$ takes $\mathbf{x}_i^r$ and $\mathbf{y}_i$ as input and outputs mean $\mu_{\mathbf{a}_i}$ and variance $\sigma_{\mathbf{a}_i}$. $E_z(*)$ and $E_a(*)$ can be neural networks on the domain we are working on because $\mathbf{x}_i^r$ may be graph structures and text reviews. For example, for graph datasets, graph convolutional neural networks could be applied. For text datasets, deep convolutional neural networks are good candidates. Then $\mathbf{z}_i$ and $\mathbf{a}_i$ can be sampled as $\mathbf{z}_i = \mu_{\mathbf{z}_i} + \sigma_{\mathbf{z}_i} \odot \epsilon_{z_i}$ and $\mathbf{a}_i = \mu_{\mathbf{a}_i} + \sigma_{\mathbf{a}_i} \odot \epsilon_{a_i}$, respectively, where ϵ_{z_i} and ϵ_{a_i} are random noises sampled from normal distributions.

Similarly, decoders are assumed as Gaussian Distribution with mean and variance as the output of neural networks:

$$\begin{aligned} p_\theta(\mathbf{x}_i^r|\mathbf{a}_i, \mathbf{z}_i) &= N(\mathbf{x}_i^r; \mu_{\mathbf{x}_i^r}, \sigma^2_{\mathbf{x}_i^r}\mathbf{I}) \quad \mu_{\mathbf{x}_i^r}, \sigma_{\mathbf{x}_i^r} = D_{x^r}(\mathbf{a}_i, \mathbf{z}_i), \\ p_\theta(\mathbf{x}_i^z|\mathbf{z}_i) &= N(\mathbf{x}_i^z; \mu_{\mathbf{x}_i^z}, \sigma^2_{\mathbf{x}_i^z}\mathbf{I}) \quad \mu_{\mathbf{x}_i^z}, \sigma_{\mathbf{x}_i^z} = D_{x^z}(\mathbf{z}_i), \\ p_\theta(\mathbf{y}_i|\mathbf{z}_i, \mathbf{a}_i) &= N(\mathbf{y}_i; \mu_{\mathbf{y}_i}, \sigma^2_{\mathbf{y}_i}\mathbf{I}) \quad \mu_{\mathbf{y}_i}, \sigma_{\mathbf{y}_i} = D_y(\mathbf{a}_i, \mathbf{z}_i), \end{aligned} \tag{14}$$

where $D_{x^r}(\cdot), D_{x^z}(\cdot), D_y(\cdot)$ are neural networks. $D_{x^r}(\cdot)$ the takes $\mathbf{a}_i$ and $\mathbf{z}_i$ as input and outputs $\mu_{\mathbf{x}_i^r}$ and $\sigma^2_{\mathbf{x}_i^r}$. Also, the input of $D_{\mathbf{x}^z}(*)$ is $\mathbf{z}_i$ and output is $\mu_{\mathbf{x}_i^z}, \sigma_{\mathbf{x}_i^z}$. Then, the input of $D_y(*)$ is $\mathbf{a}_i, \mathbf{z}_i$ and output is $\mu_{y_i}, \sigma^2_{y_i}$. And our deep learning framework is trained on Eq. (8) with mutual information loss. The overall architecture is shown in Fig. 3.

4.4. An training algorithm of FairWS

In this subsection, we will introduce the training algorithm for FairWS. The overall process is shown in Algorithm 12. The first step of our algorithm is to generate sensitive attributes based on Graphical

Probability Model in Section 4.1, which models dependency relationships between sensitive attributes, relevant features, irrelevant features and labels. Specifically, we train the encoders and decoders on labeled nodes from line 3 to line 4 on the loss Eq. (8). The implementation details are introduced in Section 4.3. Then, the latent representation of the sensitive attributes $\mathbf{A}$ is generated via line 6. Secondly, our generate latent sensitive attributes will be used to train fair classifiers by regularizing on $\mathbf{A}$ also with label loss as shown in Eq. (12). Finally, the output of this algorithm is a trained classifier and it will be used to predict labels on the testing set with unlabeled nodes.

5. Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed FairWS. Specifically, we aim to answer the following research questions:

- (**RQ1**) How does the proposed FairWS perform in terms of both classification accuracy and fairness?
- (**RQ2**) Can the proposed framework give accurate estimated sensitive attributes for achieving fairness?
- (**RQ3**) How does the quality of sensitive attributes affect the performance of the proposed FairWS?

5.1. Datasets

We conduct experiments on three publicly available benchmark datasets, including Adult [3], Credit Defaulter [1] and Animate.[1]

- **Adult:** This dataset contains records of personal yearly income. The task is to predict whether the yearly salary is over or under $50,000 and the sensitive attribute is gender. It has 12 features. We use age, relation, and marital status as relevant features $\mathbf{X}^r$ and the rest as irrelevant features.
- **Credit Defaulter:** In this dataset, each data sample is a person which has 14 features about their personal information. In addition, two samples are connected based on the similarity of their purchase and payment records, which forms a graph. The sensitive attribute of this dataset is age and the task is to classify whether a user is married. Each person's connectivity is treated as relevant features $\mathbf{X}^r$ because it is relevant to age, i.e., two persons of similar age are more likely to be connected and have similar connectivity or common friends.
- **Animate:** This dataset includes records of users' reviews and their profiles. The task is to predict whether the average ranking of users' favorite movies is in the top 400 and the sensitive attribute is whether the average scores the users give to their favorite movies are above 8. Note that the ranking of movies is evaluated from the website Animate based on their popularity and rating scores. We treat the text review from users as relevant features $\mathbf{X}^r$ and their attributes as irrelevant features. Text reviews are treated as relevant features because reviews can (i) reflect people's attitudes towards the movie, i.e. whether to give this movie a higher score; and (ii) indicate their occupations, age, or other sensitive information.

The key statistics of the datasets are summarized in Table 1, which includes the number of features for each dataset, the number of class labels, the formats of their relevant features and the number of data samples. Note that the Personal Attributes of Adult represent the feature vectors that can represent the characteristics of people. For Adult and Animate, we make the train:val:test split ratio as 5 : 2.5 : 2.5. For Credit Defaulter, following [1], we select 2000 nodes as the training set, 25% for validation and 25% for testing. Each experiment is conducted 3 times and the averaged performance will be reported.

[1] https://www.kaggle.com/marlesson/myanimelist-dataset-animes-profiles-reviews

Table 1
Statistics of datasets.

Dataset	Adult	Credit defaulter	Animate
Features	12	13	15
Class	2	2	2
Type of $\mathbf{X}^r$	Personal Attributes	Graph	Text
Data size	45,211	30,000	12,772

5.2. Experimental setup

5.2.1. Baselines

To evaluate the effectiveness of FairWS, we compare it with the vanilla model, sensitive-attribute-aware model and fair models without sensitive attributes.

- **Vanilla**: It utilizes the base classifier without the regularization form. The base classifier $g_{w_1}(\cdot)$ is MLP for all datasets. The transformed model $f_{w_2}(\cdot)$ is Graph Convolutional Network (GCN) [45] for dataset Credit Defaulter and Convolutional Neural Network (CNN) [46] for dataset Animate.
- **ConstrainS**: We assume the sensitive attribute of each sample is known for this baseline. We add the correlation regularizer between sensitive attributes and the model output to the original loss of the Vanilla model. Note that ConstrainS aims to show the accuracy and fairness we can achieve, which is the upper bound for the proposed method.

We also include following representative models in fair learning without sensitive attributes as baselines:

- **KSMOTE** [16]: It conducts clustering to get pseudo groups and treats clustering groups as pseudo sensitive attributes. It then adopts fairness regularization terms with pseudo sensitive attributes to achieve fairness.
- **RemoveR** [15]: For this baseline, it removes all candidate-relevant features for fair classifiers. This baseline is utilized to validate the efficiency of regularizing classifiers with the generated latent sensitive attributes.
- **ConstrainR**: It trains a fair classifier by calculating the correlation regularization form on relevant features through Eq. (12). We design this baseline to show the quality of our generated latent sensitive attributes in regularizing the classifier for fair predictions.
- **ARL** [14]: It optimizes the model's performance through reweighting under-represented regions detected by an adversarial model, which can alleviate bias.
- **FairRF** [15]: It uses relevant features as pseudo sensitive attributes to regularize the model to be fair. This is the state-of-the-art method to train a fair classifier without sensitive attributes.
- **SRCVAE** [30]: It is a baseline which also focuses on training fair models without sensitive attributes. It utilizes Variational Autoencoders to generate sensitive attributes and then uses generated sensitive attributes to learn a fair classifier.

For baselines FairRF and ConstrainR, since the graph and text cannot be directly utilized to regularize the model, we adopt Node2vec [47] to obtain the node embeddings as relevant features on Credit Defaulter. Similarly, we utilize the average of the pretrained word embeddings as relevant feature vectors on Animate for FairRF and ConstrainR.

5.2.2. Configurations

For ARL and KSMOTE [14,16], we utilize the authors' source codes. For other baselines, we follow the implementation of [15]. For the decoder and encoder of our sensitive estimation module, we implement them as a multi-layer perceptron (MLP) network with two and three layers, respectively. The hidden dimension is 8 for the Adult dataset, 50 for the graph dataset, and 16 for the text dataset. For classifier $g_{w_1}(\cdot)$, we adopt three-layer MLPs for the adult dataset and one-layer

Table 2
Comparison of different approaches in three datasets.

Methods	Adult			Credit defaulter			Animate		
	ACC	Δ_{EO}	Δ_{DP}	ACC	Δ_{EO}	Δ_{DP}	ACC	Δ_{EO}	Δ_{DP}
Vanilla	0.856 ± 0.001	0.046 ± 0.006	0.089 ± 0.005	0.731 ± 0.001	0.159 ± 0.001	0.101 ± 0.001	0.755 ± 0.001	0.330 ± 0.001	0.391 ± 0.001
ConstrainS	0.845 ± 0.002	0.040 ± 0.003	0.058 ± 0.004	0.713 ± 0.006	0.137 ± 0.002	0.087 ± 0.003	0.738 ± 0.008	0.202 ± 0.005	0.264 ± 0.003
ARL	0.861 ± 0.003	0.034 ± 0.012	0.141 ± 0.008	0.578 ± 0.001	**0.050 ± 0.005**	**0.054 ± 0.009**	0.688 ± 0.002	0.241 ± 0.003	0.332 ± 0.002
KSMOTE	0.560 ± 0.002	0.141 ± 0.031	0.012 ± 0.022	0.563 ± 0.003	0.203 ± 0.002	0.258 ± 0.001	0.672 ± 0.004	0.174 ± 0.001	0.320 ± 0.002
RemoveR	0.801 ± 0.010	0.124 ± 0.004	0.071 ± 0.002	0.674 ± 0.002	0.148 ± 0.003	0.092 ± 0.001	0.715 ± 0.002	0.193 ± 0.002	0.273 ± 0.002
ConstrainR	0.832 ± 0.013	0.061 ± 0.015	0.088 ± 0.019	0.668 ± 0.014	0.121 ± 0.012	0.089 ± 0.016	0.726 ± 0.017	0.257 ± 0.010	0.329 ± 0.012
FairRF	0.832 ± 0.001	0.025 ± 0.009	0.066 ± 0.004	0.682 ± 0.002	0.163 ± 0.002	0.106 ± 0.001	0.715 ± 0.002	0.225 ± 0.001	0.291 ± 0.001
SRCVAE	0.834 ± 0.002	0.038 ± 0.007	0.056 ± 0.013	0.712 ± 0.016	0.147 ± 0.009	0.089 ± 0.021	0.717 ± 0.009	0.188 ± 0.021	0.253 ± 0.017
FairWS	**0.842 ± 0.004**	0.024 ± 0.012	0.054 ± 0.010	**0.720 ± 0.012**	0.145 ± 0.016	0.087 ± 0.010	0.726 ± 0.016	**0.173 ± 0.014**	**0.247 ± 0.016**
FairWS + MI	0.833 ± 0.006	**0.013 ± 0.011**	**0.046 ± 0.019**	0.719 ± 0.028	0.145 ± 0.019	0.074 ± 0.011	**0.732 ± 0.020**	0.178 ± 0.016	0.263 ± 0.012

for other datasets. We implement two-layer GCN and CNN for graph and text datasets separately. For fair comparison, we adopt the same backbone for all baselines. Adam optimizer is adopted to train the model, with an initial learning rate of 0.001 for all datasets. We find the best hyperparameter β through $\{0.001, 0.01, 0.1, 0.5, 1, 1.5\}$ and λ through $\{0.01, 0.02, 0.03, 0.04, 0.05\}$ via grid search.

5.2.3. Evaluation metrics

For classification performance, we adopt the widely used accuracy as the evaluation metric. Following existing work on fair models [2], we adopt the difference in equal opportunity Δ_{EO} and demographic parity Δ_{DP} as the fairness metrics. They are defined as:

Equal Opportunity [2]: Equal Opportunity requires that the model assigns the equal probability of positive instances with random protected attributes i, j to a data point with a positive label:

$$\mathbb{E}(\hat{y} \mid S = i, y = 1) = \mathbb{E}(\hat{y} \mid S = j, y = 1), \tag{15}$$

where $\hat{y}$ is the output of the model g_{w_2}, S represents the sensitive attribute. Note that the tasks in our experiment are binary classification problems so y' means the probability to be predicted as positive labels. In this paper, we report the difference for equal opportunity (ΔEO), which is defined as:

$$\Delta_{EO} = |\mathbb{E}(y' \mid S = i, y = 1) - \mathbb{E}(y' \mid S = j, y = 1)|. \tag{16}$$

Demographic Parity [2]: Demographic Parity requires that the predicted results of models are fair on different sensitive groups:

$$\mathbb{E}(\hat{y} \mid S = i) = \mathbb{E}(\hat{y} \mid S = j), \forall i, j. \tag{17}$$

We also report the difference in Demographic Parity:

$$\Delta_{DP} = |\mathbb{E}(y' \mid S = i) - \mathbb{E}(y' \mid S = j)| \tag{18}$$

Note that equal opportunity and demographic parity measure fairness from different dimensions. Equal opportunity requires similar performance across protected groups; while demographic parity focuses more on fair demographics [15]. *The smaller Δ_{EO} and Δ_{DP} are, the more fair the model is.*

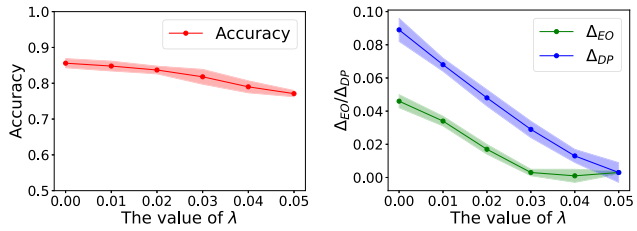
5.3. Classification accuracy and fairness

To answer **RQ1**, we conduct each experiment 3 times and report the average results along with standard deviation in terms of accuracy, Δ_{EO} and Δ_{DP} on three datasets in Table 2. Note that FairWS + MI means that we utilize the mutual information loss for the sensitive attributes estimation module and FairWS means no mutual information loss. For all baselines, the hyperparameters are tuned via grid search on the validation dataset. From Table 2, we make the following observations:

- Comparing Vanilla with ConstrainR, we can observe that directly constraining relevant features $\mathbf{X}^r$ can help the model achieve fairer results on Adult, which is because $\mathbf{X}^r$ of Adult are simple features that can be easily incorporated into the covariance regularizer. However, when $\mathbf{X}^r$ is complex features such as text on Animate and graph on Credit Defaulter, ConstrainR method does not have too much effect, which is because the learned embedding vectors obtained from an unsupervised manner, which are treated as relevant features, may be irrelevant to the targeted sensitive attributes. While for FairWS, it can extract the targeted sensitive information from the graph structure or text reviews flexibly via Variational Autoencoders in the sensitive attributes estimation part.
- Compared with baselines without the sensitive attributes, FairWS can achieve the best result in terms of the accuracy and fairness metrics on Adult datasets. Even though FairRF can achieve fairer performance on Credit Defaulter and Animate datasets separately, it will result in a significant drop in accuracy. Also, FairRF can improve the fairness performance on the Adult dataset as shown in Table 2, but it cannot work on the graph and text dataset. Finally, SRCVAE can achieve better performance by generating sensitive attributes compared with other baselines. However, they cannot outperform our model. This is because they cannot greatly eliminate the information of irrelevant features when generating sensitive attributes. Our proposed loss in Eq. (7) can discriminate sensitive attributes and irrelevant features so our model can have more accurate sensitive attributes. Based on our generated sensitive attributes, our model can obtain better performance.
- Compared with ConstrainS which uses the grand truth of sensitive attributes, FairWS can achieve comparable performance or even better performance with a little drop in accuracy for all datasets. And our proposed mutual information loss can help the fairness regularization to get better performance on the Adult dataset. In addition, it can result in comparable results on fairness metrics with higher accuracy for the text and graph dataset.
- Finally, FairWS achieve the best performance on Adult and Animate datasets in terms of accuracy and fairness metrics with different kinds of relevant features. It means that FairWS can extract sensitive information from relevant features in different formats, which proves the flexibility of our proposed model.

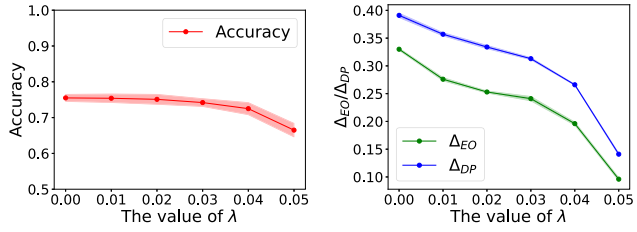
5.4. Accuracy of sensitive attribute estimation

To answer **RQ2**, we conduct an experiment to show whether our generated latent representation $\mathbf{A}$ learns information about sensitive attributes. Specifically, we adopt Gaussian Mixture Model to cluster data into two clusters based on $\mathbf{A}$ for FairWS and FairWS + MI. As $\mathbf{A}$ should contain sensitive attribute information, we would expect the two clusters would correspond to two groups of different sensitive attributes. Then, we calculate the AUC score between predicted cluster and ground sensitive attributes and the results are shown in Table 3, where Gaussian Mixture in the table means using Gaussian Mixture Model on raw relevant feature vectors. Note that this is an unsupervised setting and we evaluate the performance of the training set. From the table, we find that (i) we can obtain a large improvement compared with Gaussian Mixture on raw relevant features, which means that



(a) Accuracy (b) Δ_{EO} and Δ_{DP}

Fig. 4. Classification accuracy and fairness in terms of Δ_{EO} and Δ_{DP} w.r.t. the hyperparameter λ on Adult.



(a) Accuracy (b) Δ_{EO} and Δ_{DP}

Fig. 5. Classification accuracy and fairness in terms of Δ_{EO} and Δ_{DP} w.r.t. the hyperparameter λ on Animate.

Table 3
Comparison of different approaches for sensitive attributes estimation on three dataset.

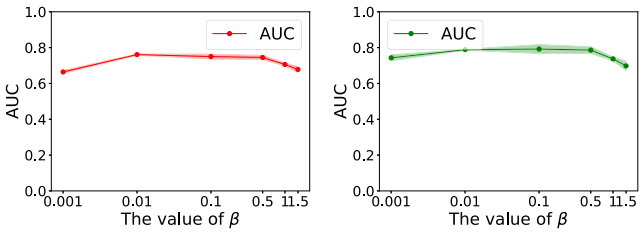
Models	Adult AUC	Credit defaulter AUC	Animate AUC
GM	0.5087 ± 0.012	0.5238 ± 0.005	0.5363 ± 0.009
SRCVAE	0.7021 ± 0.028	0.6741 ± 0.019	0.6862 ± 0.002
FairWS	0.7481 ± 0.010	**0.7046 ± 0.036**	0.7804 ± 0.003
FairWS+MI	**0.7704 ± 0.046**	0.6994 ± 0.004	**0.7926 ± 0.001**

FairWS can efficiently estimate sensitive attributes via extracting sensitive information from relevant features; and (ii) our model FairWS can consistently outperform SRVAE. It verifies the effectiveness of our model to estimate sensitive attributes. Furthermore, FairWS+MI can outperform SRCVAE and FairWS on Adult and Animate datasets. It shows that mutual information loss can help **A** learn more information about sensitive attributes. It demonstrates that disentangling latent representation between $\mathbf{a}_i$ and $\mathbf{z}_i$ can help $\mathbf{a}_i$ to better learn sensitive information.

5.5. Hyperparameter sensitivity analysis

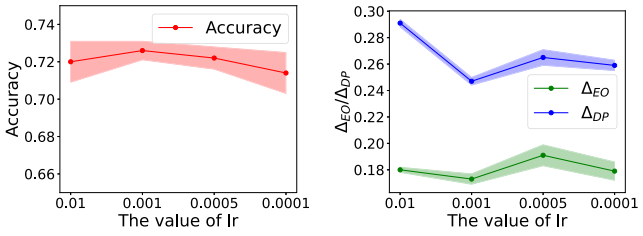
The proposed FairWS has two important hyperparameters λ and β. λ controls the trade-off between fairness and accuracy when learning the fair classifier. To evaluate the parameter sensitivity on λ, we fix the sensitive attributes estimation module with $\beta = 0.01$ and train a fair classifier based on the generated latent representation with different λ. We vary λ as $\{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$. Fig. 4 and 5 show the results on Adult and Animate datasets separately. From Fig. 4, we can observe that larger λ will lead to a slight drop in terms of accuracy but significant improvement in terms of fairness Δ_{EO} and Δ_{DP}, which is because the higher weight of correlation loss between predicted results and sensitive information will lead to a fairer model but with a drop of accuracy. For Fig. 5 which has text relevant features, and also shows the same pattern as Fig. 4. Thus, it is important to select a λ for various requirements, e.g., better performance on the accuracy or fairness metrics.

Another hyperparameter β is to control the training process of the latent representation **A**. We vary β as $\{0.001, 0.01, 0.1, 0.5, 1, 1.5\}$. For each choice of β, we learn **A** and use Gaussian Mixture Model to predict the sensitive attributes from **A**. The results of sensitive attribute estimation in terms of AUC are shown in Fig. 6. We observe that the performance first increases when β increases from 0.001 to 0.01, and then results tend to be fluent on Adult and Animate. Finally, when β is larger than 0.5, the AUC values will have a slight drop. It is because higher β will make $q_\phi(\mathbf{a}_i \mid \mathbf{x}_i^r, \mathbf{y}_i)$ close to standard Gaussian distributions, which will fail to extract sensitive information from $\mathbf{x}_i^r$. Therefore, small and large β will result in less sensitive information in **A** and the best value of it is between 0.01 to 0.5.



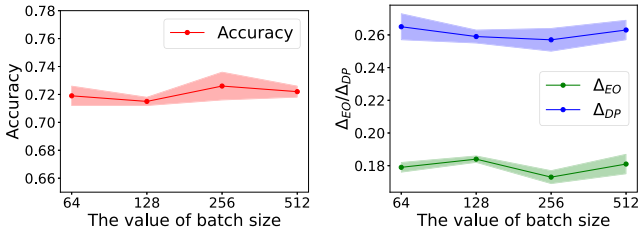
(a) Adult (b) Animate

Fig. 6. Sensitivity of β on sensitive attribute estimation.



(a) Accuracy (b) Δ_{EO} and Δ_{DP}

Fig. 7. Classification accuracy and fairness in terms of Δ_{EO} and Δ_{DP} w.r.t. the hyperparameter learning rate on the Animate dataset, which is denoted as lr in the figure.



(a) Accuracy (b) Δ_{EO} and Δ_{DP}

Fig. 8. Classification accuracy and fairness in terms of Δ_{EO} and Δ_{DP} w.r.t. the hyperparameter batch size on the Animate dataset.

We also analyze the influence of batch size and the learning rate of our model. We vary the learning rate as {0.01, 0.001, 0.0005, 0.0001}. Other hyperparameters are determined by cross validation with grid search. The corresponding results are shown in Fig. 7. We can observe that a higher learning rate can result in unstable training and poor performance because it causes the model to take larger steps during parameter updates. This can lead to overshooting the optimal parameter values and oscillations in the loss function. The model may fail to converge or converge to a suboptimal solution. A lower learning rate can also make the model more susceptible to getting trapped in local minima or plateaus in the loss landscape. Since the parameter updates are small, the model may struggle to escape these regions and find the global or better local optima.

We also vary the batch size as {64, 128, 256, 512}. Other hyperparameters are determined by cross validation with grid search. The

Table 4
The impact of the quality of relevant features on the sensitive attribute estimation on Adult.

Methods	Random	Top-1	Noisy	GM	FairWS
AUC	0.6681 ± 0.019	0.6678 ± 0.022	0.7019 ± 0.027	0.5078 ± 0.018	0.7616 ± 0.025

Table 5
Comparison of different selection approaches on relevant features.

Methods	ACC	Δ_{EO}	Δ_{DP}
Vanilla	0.856 ± 0.001	0.046 ± 0.006	0.089 ± 0.005
Random	0.826 ± 0.020	0.036 ± 0.015	0.057 ± 0.014
Top-1	0.841 ± 0.011	0.041 ± 0.008	0.057 ± 0.010
Nosiy	0.838 ± 0.012	0.031 ± 0.021	0.059 ± 0.021
FairWS	0.842 ± 0.004	0.024 ± 0.012	0.054 ± 0.010

corresponding results are shown in Fig. 8. We can find that batch size may not have too much influence on the final results because the presence of regularization techniques, such as dropout or weight decay, can further mitigate the influence of batch size on the final results by introducing robustness to the training process [48,49].

5.6. Impact of relevant features

In this section, to answer **RQ3**, we explore the impact of the quality of relevant features on FiarWS. To get relevant features of different quality, we consider the following variants of FairWS:

- **Random**: We randomly select a set of relevant features with the same number of attributes with FairWS.
- **Top-1**: It includes the most-effective relevant features. We test all candidate relevant features and select the one which achieves the highest performance by regularizing a classifier based on Eq. (10), and report its performance.
- **Noisy**: It contains features randomly selected from both highly relevant features and irrelevant features. In implementation, we replace one attribute in the highly relevant features e.g. age, relation and marital status with one irrelevant features.

We first evaluate the quality of **A** under different choices of relevant features and report the results in Table 4. In the table, GM means applying Gaussian Mixture on raw relevant features and we consider it as the baseline or reference result for our analysis. For a fair comparison, Random, Noisy and Top-1 are three selection methods for relevant features and we use them to select three relevant features. Then, we utilize the selected relevant features from these methods to train the FairWS model. We only conduct an experiment on Adult because Adult is the only dataset which has tabular attributes with clear semantic meaning and requires our prior knowledge to select relevant features; while the other two datasets have texts and features, which make it difficult to control the experiment. The results on Adult are shown in Table 4. From the table, we can observe that FairwS can also learn information about the sensitive attributes even with noisy relevant features. Comparing FairWS with Top-1, we can find that sensitive information in one feature is limited but FairWS can utilize a set of relevant features to learn sensitive information automatically and achieve great performance.

Furthermore, we conduct experiments to explore the impact of relevant features in terms of accuracy and fairness metrics. The results are shown in Table 5. We make the following observation:

- Firstly, comparing Noisy with FairWS, with noisy relevant features where we randomly replace one highly relevant features with other features, our model can help to train a fair classifier with a little drop in accuracy. Also, in comparison with **Random**, training our models based on a random selection of features can efficiently extract sensitive information to regularize the MLP classifier. It can achieve similar results on fairness metrics with more drops in accuracy. It shows that FairRF can cope with little domain knowledge scenarios.
- In comparison with Top-1, FairWS still shows great improvement. It further proves the ability of FairWS to extract sensitive information from a set of relevant features and sensitive information in one feature is limited.

6. Conclusion

In this paper, we study a novel problem of training fair and accurate classifiers without sensitive attributes by estimating sensitive information from features which are relevant to sensitive attributes. We propose a novel framework FairWS which learns sensitive information from relevant features and regularizes classifiers based on inferred sensitive information. FairWS can flexibly learn sensitive information from relevant features in different formats and even from noisy relevant features which may contain irrelevant features. Through extensive experiments, we demonstrate that our method significantly outperformed the state-of-the-art methods w.r.t both accuracy and fairness metrics when sensitive attributes are unavailable. Also, we explore the impact of relevant features which proves FairWS can obtain sensitive information even with irrelevant features. Parameter sensitive analysis is also conducted to understand the sensitivity to hyperparameters. In the future, work can be done to adopt our generated sensitive information to more fair models. Furthermore, designing fair models without sensitive attributes on different kinds of data is also a promising direction, including graphs, text and images. Finally, it is also significant to explore training fair models without any prior knowledge.

CRediT authorship contribution statement

Huaisheng Zhu: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Visualization. **Enyan Dai:** Conceptualization, Methodology, Writing – review & editing. **Hui Liu:** Conceptualization, Methodology, Writing – review & editing. **Suhang Wang:** Conceptualization, Methodology, Writing – review & editing, Investigation, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Suhang Wang reports financial support was provided by National Science Foundation. Suhang Wang reports financial support was provided by Army Research Office.

Data availability

Data will be made available on request.

Acknowledgments

This material is based upon work supported by, or in part by, the National Science Foundation (NSF) under grant #IIS-1909702, and Army Research Office (ARO) under grant #W911NF21-1-0198. The findings and conclusions in this paper do not necessarily reflect the view of the funding agency.

References

[1] Y. Dong, N. Liu, B. Jalaian, J. Li, EDITS: Modeling and mitigating data bias for graph neural networks, 2021, arXiv preprint arXiv:2108.05233.

[2] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Comput. Surv. 54 (6) (2021) 1–35.

[3] A. Asuncion, D. Newman, UCI Machine Learning Repository, Irvine, CA, USA, 2007.

[4] A. Beutel, J. Chen, Z. Zhao, E.H. Chi, Data decisions and theoretical implications when adversarially learning fair representations, 2017, arXiv preprint arXiv:1707.00075.

[5] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, Adv. Neural Inf. Process. Syst. 29 (2016) 3315–3323.

[6] B.H. Zhang, B. Lemoine, M. Mitchell, Mitigating unwanted biases with adversarial learning, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 335–340.

[7] M.B. Zafar, I. Valera, M.G. Rogriguez, K.P. Gummadi, Fairness constraints: Mechanisms for fair classification, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 962–970.

[8] M. Feldman, S.A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian, Certifying and removing disparate impact, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 259–268.

[9] D. Xu, S. Yuan, L. Zhang, X. Wu, Fairgan: Fairness-aware generative adversarial networks, in: 2018 IEEE International Conference on Big Data, Big Data, IEEE, 2018, pp. 570–575.

[10] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, O. Bachem, On the fairness of disentangled representations, 2019, arXiv preprint arXiv:1905.13662.

[11] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012.

[12] G. Pleiss, M. Raghavan, F. Wu, J.M. Kleinberg, K.Q. Weinberger, On fairness and calibration, in: NIPS, 2017.

[13] A. Coston, K.N. Ramamurthy, D. Wei, K.R. Varshney, S. Speakman, Z. Mustahsan, S. Chakraborty, Fair transfer learning with missing protected attributes, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 91–98.

[14] P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, E.H. Chi, Fairness without demographics through adversarially reweighted learning, 2020, arXiv preprint arXiv:2006.13114.

[15] T. Zhao, E. Dai, K. Shu, S. Wang, You can still achieve fairness without sensitive attributes: Exploring biases in non-sensitive features, 2021, arXiv preprint arXiv:2104.14537.

[16] S. Yan, H.-t. Kao, E. Ferrara, Fair class balancing: enhancing model fairness without observing sensitive attributes, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 1715–1724.

[17] P. Wang, J. Guo, Y. Lan, J. Xu, X. Cheng, Multi-task representation learning for demographic prediction, in: European Conference on Information Retrieval, Springer, 2016, pp. 88–99.

[18] J. Otterbacher, Inferring gender of movie reviewers: exploiting writing style, content and metadata, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, 2010, pp. 369–378.

[19] D.V. Phuong, T.M. Phuong, Gender prediction using browsing history, in: Knowledge and Systems Engineering, Springer, 2014, pp. 271–283.

[20] J. Kang, J. He, R. Maciejewski, H. Tong, Inform: Individual fairness on graph mining, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 379–389.

[21] P. Lahoti, K.P. Gummadi, G. Weikum, Operationalizing individual fairness with pairwise fair representations, 2019, arXiv preprint arXiv:1907.01439.

[22] A.J. Biega, K.P. Gummadi, G. Weikum, Equity of attention: Amortizing individual fairness in rankings, in: The 41st International ACM Sigir Conference on Research & Development in Information Retrieval, 2018, pp. 405–414.

[23] L. Zhang, Y. Wu, X. Wu, Achieving non-discrimination in data release, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 1335–1344.

[24] C. Zhang, J.A. Shah, Fairness in multi-agent sequential decision-making, Adv. Neural Inf. Process. Syst. 27 (2014).

[25] T. Hashimoto, M. Srivastava, H. Namkoong, P. Liang, Fairness without demographics in repeated loss minimization, in: International Conference on Machine Learning, PMLR, 2018, pp. 1929–1938.

[26] M. Mohri, G. Sivek, A.T. Suresh, Agnostic federated learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 4615–4625.

[27] K.N. Vokinger, S. Feuerriegel, A.S. Kesselheim, Mitigating bias in machine learning for medicine, Commun. Med. 1 (1) (2021) 25.

[28] C. Hurlin, C. Pérignon, S. Saurin, The fairness of credit scoring models, 2022, arXiv preprint arXiv:2205.10200.

[29] R. Berk, H. Heidari, S. Jabbari, M. Kearns, A. Roth, Fairness in criminal justice risk assessments: The state of the art, Sociol. Methods Res. 50 (1) (2021) 3–44.

[30] V. Grari, S. Lamprier, M. Detyniecki, Fairness without the sensitive attribute via causal variational autoencoder, 2021, arXiv preprint arXiv:2109.04999.

[31] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.

[32] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: An overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.

[33] Z. Hu, Z. Yang, R. Salakhutdinov, E.P. Xing, On unifying deep generative models, 2017, arXiv preprint arXiv:1706.00550.

[34] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, E.P. Xing, Toward controlled generation of text, in: International Conference on Machine Learning, PMLR, 2017, pp. 1587–1596.

[35] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, Adv. Neural Inf. Process. Syst. 28 (2015).

[36] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, in: International Conference on Machine Learning, PMLR, 2017, pp. 2642–2651.

[37] S.R. Bowman, L. Vilnis, O. Vinyals, A.M. Dai, R. Jozefowicz, S. Bengio, Generating sentences from a continuous space, 2015, arXiv preprint arXiv:1511.06349.

[38] A. Siarohin, E. Sangineto, S. Lathuiliere, N. Sebe, Deformable gans for pose-based human image generation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3408–3416.

[39] E. Creager, D. Madras, J.-H. Jacobsen, M. Weis, K. Swersky, T. Pitassi, R. Zemel, Flexibly fair representation learning by disentanglement, in: International Conference on Machine Learning, PMLR, 2019, pp. 1436–1445.

[40] C. Louizos, K. Swersky, Y. Li, M. Welling, R. Zemel, The variational fair autoencoder, 2015, arXiv preprint arXiv:1511.00830.

[41] A. Amini, A.P. Soleimany, W. Schwarting, S.N. Bhatia, D. Rus, Uncovering and mitigating algorithmic bias through learned latent structure, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 289–295.

[42] D. Moyer, S. Gao, R. Brekelmans, A. Galstyan, G. Ver Steeg, Invariant representations without adversarial training, Adv. Neural Inf. Process. Syst. 31 (2018).

[43] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, beta-vae: Learning basic visual concepts with a constrained variational framework, 2016.

[44] M.I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, R.D. Hjelm, Mine: mutual information neural estimation, 2018, arXiv preprint arXiv:1801.04062.

[45] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[46] S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back, Face recognition: A convolutional neural-network approach, IEEE Trans. Neural Netw. 8 (1) (1997) 98–113.

[47] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.

[48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[49] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017, arXiv preprint arXiv:1711.05101.

Huaisheng Zhu received the B.E. degree in Computer Science and Engineering from Southern University of Science and Technology. He is currently working towards his Ph.D. degree at The Pennsylvania State University under the supervision of Professor Suhang Wang. His research interests are: data mining, graph neural networks, and machine learning.

Enyan Dai is currently working towards his Ph.D. degree at The Pennsylvania State University under the supervision of Professor Suhang Wang. Before that, he received the B.S. degree in mechanical engineering from University of Science and Technology of China, and M.S. degree in AI from KU Leuven. His research interests are: data mining, graph neural networks, and trustworthy AI. He has published innovative works in top conference proceedings such as ICLR, WSDM, KDD, WWW, CIKM, and ICWSM.

Hui Liu is an Assistant Professor in the computer science and engineering department at Michigan State University. She received her Ph.D. degree in Electrical Engineering from Southern Methodist University in 2015. Her research interests include trustworthy AI, designing data mining algorithms for wireless communication of smart devices, and applying machine learning and data mining in wireless communications.

Suhang Wang is an assistant professor of the College of Information Sciences and Technology, The Pennsylvania State University. He received his Ph.D. in Computer Science from Arizona State University in 2018, M.S. in Electrical Engineering from University of Michigan - Ann Arbor in 2013, and B.S. in Electrical and Computer Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012. His research interests are in graph mining, data mining and machine learning. He is an associate editor for several journals and serves as regular journal reviewers and numerous conference program committees. He has published innovative works in highly ranked journals and top conference proceedings such as IEEE TKDE, ACM TIST, KDD, WWW, AAAI, IJCAI, CIKM, SDM, WSDM, ICDM and CVPR, which have received extensive coverage in the media.