Fair Heterogeneous Network Embeddings

(Draft Version, Accepted by ICWSM 2021)

Abstract

Recently, much attention has been paid to the societal impact of AI, especially concerns regarding its fairness. A growing body of research has identified unfair AI systems and proposed methods to debias them, yet many challenges remain. Heterogeneous Network Embedding (HNE), a popular technology used in complex network mining, has socially consequential applications such as automated career counseling, but there have been few attempts to ensure that it will not encode or amplify harmful biases, e.g. sexism in the job market. To address this gap, in this paper we propose a comprehensive set of debiasing methods for fair HNE, including sampling-based, projection-based, and graph neural network-based techniques. We systematically study the behavior of these algorithms, especially their capability in balancing the trade-off between fairness and prediction accuracy. We evaluate the performance of the proposed methods in an automated career counseling application where we mitigate gender bias in career recommendation. Based on the evaluation results on two datasets, we identify the most effective fair HNE techniques under different conditions.

Introduction

Many real-world social and information networks such as social networks, bibliographic networks and biological networks are heterogeneous in nature (Liu et al. 2016; Zhou et al. 2007; Xiong et al. 2019). We can model such networks as Heterogeneous Information Networks (HINs) which contain diverse types of nodes and/or relationships. For example, we can model the Twitter social network as an HIN where the nodes are users or tweets and the links are the follower/following relations between users and the authoring/retweeting relations between a user and a tweet.

To support large-scale heterogeneous network mining, much attention has recently been paid to representation learning where each node in a network is automatically mapped to a dense vector in a low dimensional embedding space which preserves the relationships between nodes and important structural characteristics of the original networks (Fu, Lee, and Lei 2017; Dong, Chawla, and Swami 2017). Due to its robustness, flexibility and scalability, heterogeneous network embedding (HNE) has been widely used to support diverse network mining tasks such as node classification (Dong, Chawla, and Swami 2017), link prediction (Wang et al. 2018), community detection (Cavallari et al. 2017), and recommender systems (Shi et al. 2019).

Despite its popularity, little attention has yet been paid to the understanding and mitigation of biases toward certain demographics in HNE, such as gender and racial biases. As demonstrated in a wide range of recent discoveries, machine learning systems trained with human-generated content (e.g., social media data) frequently inherit or even amplify human biases in the data (Dastin 2018; Noble 2018; Angwin et al. 2016). For example, word embedding models, which inspired some of the early work on network embedding such as DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), were shown to exhibit female/male gender stereotypes to a disturbing extent (e.g., "man is to computer programmer as woman is to homemaker") (Bolukbasi et al. 2016). To overcome this, there has been a concentrated recent effort in the natural language processing community (Bolukbasi et al. 2016; Caliskan, Bryson, and Narayanan 2017; Gonen and Goldberg 2019) on understanding and mitigating the biases in word embeddings. In the network mining community however, not much attention has been paid to the biases in HNE. Since HNE may encode harmful societal prejudice, it may cause unintended bias or unfairness in downstream applications. Therefore, it is important that we make sure HNE are unbiased and applications incorporating these embeddings are fair and will not negatively impact vulnerable people and marginalized communities in our society.

In this paper, we propose a range of fair HNE algorithms, including sampling-based, projection-based, and graph neural network-based methods to mitigate demographic bias in HNE. We systematically study the behavior of these algorithms, especially their capability in balancing the trade-off between prediction accuracy and fairness.

To evaluate our algorithms, we applied our fair HNE techniques to automated fair career recommendation. Career counseling plays an important role in many people's lives. Unbiased career advice based on an accurate assessment of one's interests, skills and personality can help them make proper career choices. Good career choice in turn can boost their economic success, social standing, and quality of life. Biased career counseling however may restrict occupational opportunities and stunt the career development of disadvantaged populations (e.g., girls and minorities) (Alshabani and Soto 2020).

The main contributions of this work include:

- To the best of our knowledge, this is the first systematic investigation on measuring and mitigating demographic bias in heterogeneous information networks. Although prior work has studied mitigating bias in network embedding (Rahman et al. 2019), they focused on homogeneous instead of heterogeneous networks.
- We propose a comprehensive suite of de-biasing algorithms ranging from sampling-based, projection-based, to graph neural network-based techniques to mitigate demographic biases in HNE.
- We demonstrate the effectiveness of the proposed methods in mitigating gender bias in automated career counseling on two real world datasets. Our results illuminate the prediction accuracy vs. fairness trade-off behavior of these algorithms, providing guidance to practitioners.

The rest of the paper is organized as follows. We start with a brief survey of related work, followed by a problem statement and the details of the proposed fair HNE algorithms. In the next section, we discuss the experiments designed to evaluate the effectiveness of the proposed methods under various conditions. We conclude the paper by summarizing the main findings and pointing out some future directions.

Related Work

In this section we survey the research areas relevant to our paper. First, we summarize the main heterogeneous network embeddings methods since they are the basis of our debiasing algorithms. Next, we survey the general area of fair machine learning with a special focus on two most relevant subareas: fair word embeddings and fair homogeneous network embeddings.

Heterogeneous Network Embeddings (HNE)

We categorize the typical HNE methods into two main types. *Task-agnostic HNE* methods focus on training a general-purpose network embedding that can be used to support different network mining tasks. In contrast, *task-specific supervised HNE* methods focus on training network embeddings that are optimized for a particular final task.

Task-agnostic HNE Task-agnostic HNE methods normally employ a meta-path guided sampling method to generate paths. After obtaining meta-paths, typical word embedding algorithms such as word2vec (Mikolov et al. 2013a; 2013b) can be employed on the instantiated meta-path sequences to learn HNE. A meta-path is a path consisting of a sequence of relations defined between different object types, which is either specified manually or derived from additional supervision (Fu, Lee, and Lei 2017; Dong, Chawla, and Swami 2017; Shi et al. 2018). For example, a meta-path $\texttt{author} \xrightarrow{\texttt{write}} \texttt{paper} \xrightarrow{\texttt{written by}} \texttt{author}$ in a bibliographic network represents a co-authorship relation in a paper.

Task-specific Supervised HNE Graph Neural Networks (GNNs) can learn HNE with supervision by translating nodes of a specific type into labels. GNN-based methods are end-to-end supervised approaches to learn network embeddings. GNNs have the ability to aggregate local features, and to learn highly expressive representations. A comprehensive survey on GNNs (Wu et al. 2020) shows that the majority of the current GNNs are designed for homogeneous networks. Recently, (Zhang et al. 2018) proposed a GNN method to explore heterogeneous networks. They translated an HIN into multiple homogeneous networks and applied GNNs to each homogeneous network and then combined them at the final layer.

Fair Machine Learning: General Approaches

There is a rising awareness that bias and fairness issues in machine learning (ML) algorithms can cause substantial societal harm (Angwin et al. 2016; Buolamwini and Gebru 2018). A concentrated effort in the machine learning community aims to address this problem. Existing methods on fair machine learning can be summarized into three general strategies.

The first strategy employs fairness-aware pre-processing to adapt the training data, including (a) modifying the values of sensitive attributes and class labels; (b) mitigating the dependencies between sensitive attributes and class labels by mapping the training data to another space (Dwork et al. 2012a; Feldman et al. 2015), and (c) learning a fair representation of the training data that is independent of the protected attribute (Zemel et al. 2013; Xie et al. 2017).

The second strategy focuses on employing a fairness-guided optimization in model training. A fairness objective is added as a constraint or a regularization term to the existing optimization objective to enforce fairness. (Calders and Verwer 2010; Zafar et al. 2017b; 2017a). For example, (Zafar et al. 2017b) minimizes the covariance between the sensitive attributes and the (signed) distance between feature vectors and the decision boundary of a classifier.

The third strategy modifies posteriors to satisfy fairness constraints. For example, (Hardt, Price, and Srebro 2016) selected a threshold such that the true positive rates of different groups are equal.

Fair Word Embeddings

Word embedding models (Mikolov et al. 2013b) learn a mapping of each word a text vocabulary to a vector in a embedding space to encode semantic meaning and syntactic structures of natural languages. These models are typically trained using a neural network-based representation learning algorithm on word co-occurrence data computed from massive text corpora. Since text data may contain societal stereotypes such as racism or sexism, word embeddings also typically inherit or amplify biases present in the data (Bolukbasi et al. 2016; Caliskan, Bryson, and Narayanan 2017; Papakyriakopoulos et al. 2020).

The most popular method of debiasing word embeddings is to project each word embedding orthogonally to the bias direction (Bolukbasi et al. 2016) followed by a crowdsourcing based location correction.

Fair Homogeneous Network Embedding

Fair network embedding is a relatively new area which is now beginning to be addressed. In an approach related to ours, (Rahman et al. 2019) modified random walks to learn fairness aware embeddings. At each step, the algorithm partitions neighbors into different groups based on the values of the sensitive attribute. The system tries to give each group the same probability of being selected regardless of its size. The method is applicable only to homogeneous networks, while this work addresses heterogeneous networks.

Problem Statement

We first describe our problem setting. We assume that our dataset is a heterogeneous information network $G = (V, E, T, R)$ with $|T| > 1$, where V denotes a set of nodes, E denotes a set of edges, T denotes a set of node types (e.g., user, career, and item), and R denotes a set of edge relations (e.g., a user likes a Facebook page, a user rates a movie, and a user chooses a career). We also assume a binary protected attribute a, which pertains to nodes of at least one type (e.g. gender for the users). An illustrative example of a heterogeneous Facebook career network is shown in Figure 1.

We focus on fairness in the general task of link prediction in heterogeneous information networks, which we illustrate with an application to career counseling. Our goal is to learn node embeddings e_v for all nodes $v \in V$ such that their use for link prediction on held-out edges leads to accurate performance, and the system behaves in a fair manner with regard to the protected demographics a. Fairness is a complex socio-technical issue. So far many fairness definitions have been proposed in the AI community. Here we adopt two of the most widely used AI fairness definitions and consider their application to link prediction or node classification in heterogeneous networks, e.g. for career recommendation.

Demographic Parity Demographic Parity is one of the most well-known criteria for fairness in machine learning classifiers (Dwork et al. 2012b). It is defined as:

$$P(\hat{y} = k|a = 0) = P(\hat{y} = k|a = 1)\ \forall k,$$

where $\hat{y}$ is a predicted label, y is the ground truth class label, a is the protected attribute such as gender (here we only consider binary protected attributes), and k represents the possible values of the class label y. The extent to which the demographic parity criterion is violated is measured by a distance metric between the two conditional distributions, typically chosen to be the total variation distance (see Eq. 9).

In the case of link prediction in heterogeneous networks, we adapt the definition by letting $P(\hat{y}|a)$ be the empirical probability of a link between nodes of protected type a (e.g. female users) and a sensitive node y (e.g. a career recommendation, such as engineer, doctor, etc). Based on this definition, in career counseling, to achieve perfect demographic parity, the probability of recommending a specific career (e.g, computer science) given a male should be the same as that given a female.

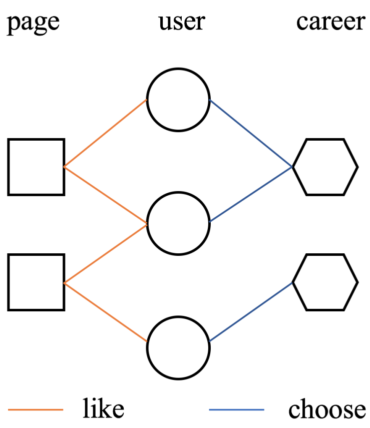


Figure 1: An illustrative example of a Facebook career network. Different shapes (colors) indicate different types of nodes (relationships). Orange line represents “like” relationship. Blue line represents “choose” relationship.

Equal Opportunity Another popular fairness definition is *equal opportunity* (Hardt, Price, and Srebro 2016). It is defined as:

$$P(\hat{y} = k|a = 0, y = k) = P(\hat{y} = k|a = 1, y = k)\ \forall k\ .$$

We adapt this to link prediction in heterogeneous networks analogously to demographic parity. Based on this definition, in career counseling, to achieve perfect equal opportunity, for a subgroup of people who indeed has chosen a particular career (e.g., in computer science), the probability of the system to recommend this career given a person is a male should be the same as that given the person is a female.

Methodology

In this section, we present three classes of fair HIN embedding methods: (1) *fairness-aware sampling* which extends existing sampling methods for HIN embedding to mitigate bias, (2) *embedding projection* which employs a vector-space projection operation to reduce biases in the embeddings, and (3) *supervised learning with a fairness objective*, which employs a graph neural network (GNN) to learn task-specific network embeddings that optimize both prediction accuracy and fairness.

To illustrate these algorithms, we use fair career counseling as an example application. We formulate fair career counseling as a fair link prediction in HIN problem. Here, the nodes in the HIN includes a *user* (e.g., a Facebook user or a movie reviewer), an *item* (e.g., a Facebook page or a movie) and a *career* (e.g., the *career concentration* or *occupation* declared by a user). A user node is linked to an item node via a “like” link, indicating a user’s preference for the item. A user node can also be linked to a “career” node via a “choose” link. The goal of automated career recommendation is to predict whether there should be a link between a user and a career based on the interests/likes of a user. In the following, we propose a range of methods for fair HNE.

Fairness-Aware Sampling

To learn embeddings in a heterogeneous network, we often use meta-paths to guide a random walker to generate traversal paths. A meta-path is a path consisting of a sequence of relations defined between different node types. For example, $\texttt{user} \xrightarrow{\texttt{choose}} \texttt{career} \xrightarrow{\texttt{chosen by}} \texttt{user}$ represents two users choosing the same career. After obtaining meta-paths, we can use a meta-path based heterogeneous network representation learning algorithm such as metapath2vec (M2V) to learn the node embeddings. Specifically, the objective function of M2V is defined as follows,

$$\underset{\theta}{\operatorname{argmax}} \sum_{v \in V} \sum_{t \in T} \sum_{c_t \in N_t(v)} \log p(c_t | v; \theta), \tag{1}$$

where c_t is the context node of type t, $N_t(v)$ denotes node v’s neighbors which are of type t, and $p(c_t|v;\theta)$ is commonly defined as a softmax function. Here the main difference between a context node and a neighboring node is that a context node is a node within a sliding windows along a path while neighboring nodes are the k-hop nodes from a center node. The objective function of M2V is similar to that of word2vec except that M2V is sensitive to the type of a node.

After learning user embeddings and career embeddings, we then train a multilayer perceptron (MLP) to predict the career, where the input is a user embedding and the career embeddings learned by M2V.

Traditional Meta-path Generation A meta-path $\mathcal{P}$ is a path defined on the graph $G = (V, E, T, R)$, and is denoted in the form of $V_1 \xrightarrow{R_1} V_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} V_{l+1}$. This meta-path defines a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between type 1 and type $l + 1$, where $\circ$ denotes the composition operator on relations, and V_t denotes all nodes with type t.

Here we show how to use meta-paths to guide heterogeneous random walkers. The transition probability at step i is defined as follows,

$$P(v^{i+1}|v_t^i) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|}, & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0, & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0, & (v^{i+1}, v_t^i) \notin E \end{cases}, \tag{2}$$

where $v_t^i \in V_t$, and $N_{t+1}(v_t^i)$ denotes v_t^i’s neighbors which are of type $t + 1$, and $\phi(\cdot)$ is a function returning the type of a node.

We manually define two meta-paths for our career counseling application. $\texttt{career} \xrightarrow{\texttt{chosen by}} \texttt{user} \xrightarrow{\texttt{like}} \texttt{item} \xrightarrow{\texttt{liked by}} \texttt{user} \xrightarrow{\texttt{choose}} \texttt{career}$ represents two careers chosen by two different users who like the same item (e.g., a Facebook page, or a movie). $\texttt{user} \xrightarrow{\texttt{like}} \texttt{item} \xrightarrow{\texttt{liked by}} \texttt{user}$ represents two users like the same item.

Fair Meta-path Generation Assume a is a protected attribute and i is a value of a, g_i represents a group of users satisfying $a = i$. If a is a binary variable. We define g_i is an advantaged group if it includes a larger number of training examples than the other group (the disadvantaged group). Fair meta-path generation aims to up-sample the disadvantaged group with higher probability while down-sample the advantaged group. The sampling probability is inversely proportional to the number of users in each group. As the number of users in the disadvantaged group is less than that in the advantaged group, the disadvantaged group has higher probability to be sampled. As a result, the system is less likely to neglect the disadvantaged group.

Next, we present the details of the algorithm. Fair meta-path sampling occurs at the step where the current node type is *career* (C) and the next node type is *user* (U). Assume a is a protected attribute of a user (e.g., gender or race). To simplify our explanation, we assume a is binary. Based on the values of a, we can cluster all the users into different groups (e.g., g_0 and g_1). The unnormalized transition probability at step i where $t = c$ and $t + 1 = u$ is defined as follows:

$$P(v^{i+1}|v_C^i) = \begin{cases} \frac{r}{|N(v_C^i, U, g_0)|}, & cond, \pi(v^{i+1}) = g_0 \\ \frac{r}{|N(v_C^i, U, g_1)|}, & cond, \pi(v^{i+1}) = g_1 \\ 0, & (v^{i+1}, v_C^i) \in E, \phi(v^{i+1}) \neq U \\ 0, & (v^{i+1}, v_C^i) \notin E, \end{cases} \tag{3}$$

where $cond$ is the condition where $(v^{i+1}, v_C^i) \in E, \phi(v^{i+1}) = U$, $v_C^i \in V_C$, which means the current node is a career node and the next node is a user node, and they are connected, and $N(v_C^i, U, g_0)$ denotes the neighbors of v_C^i which are of type U and belong to group g_0, and $N(v_C^i, U, g_1)$ denotes the neighbors of v_C^i’ which are of type U and belong to group g_1, and $\phi(\cdot)$ is a function returning the type of a node, and $\pi(\cdot)$ is a function returning the value of the protected attribute(e.g., the gender of a user), and r is a hyper-parameter that can determine to what extend the random walker over samples the disadvantaged group in order to correct bias. Note that if $|N(v_C^i, U, g_0)| = 0$ or $|N(v_C^i, U, g_1)| = 0$, then the corresponding unnormalized probability should be zero. The transition probability in other conditions remains the same as that in Eq. 2.

Embedding Projection

The second debiasing method is inspired by a recent work on attenuating bias in word vectors (Dev and Phillips 2019). It is often used as a post-processing step. We adapt this method to debias HIN embeddings. Basically, after obtaining network embeddings using any HIN embedding methods, we can reduce their unfairness by eliminating the effect of a protected attribute (e.g., gender) in learned user embeddings via vector projection. The main difference between our method and the projection method used in (Dev and Phillips 2019) is the computing of the bias direction in the embedding space.

(Dev and Phillips 2019) computed a "bias direction" for word embeddings based on the difference in the average embeddings of male and female names. In our case, let v_{g_i} be the direction of group g_i. We compute v_{g_i} by averaging all the embeddings of the users in group g_i,

$$v_{g_i} = \frac{1}{n_i} \frac{e_{u_1} + e_{u_2} + \cdots + e_{u_{n_i}}}{\|e_{u_1} + e_{u_2} + \cdots + e_{u_{n_i}}\|}, \qquad (4)$$

where e_{u_1}, e_{u_2}, $e_{u_{n_i}}$ are the user embeddings of the individuals in g_i, and there are n_i users in group g_i. Assuming the protected attribute is binary, we can compute the bias direction v_b using

$$v_b = v_{g_0} - v_{g_1} \ . \qquad (5)$$

.

To reduce bias in user embeddings, we project each users vector e_u orthogonally to the bias direction v_b to obtain the "bias component" of the embedding, which we subtract:

$$e'_u = e_u - < e_u, v_b > v_b, \qquad (6)$$

where v_b is the bias direction, $<, >$ is the inner product operation, and e'_u is the resulting debiased user embedding.

Fairness-Aware Graph Neural Network-based Learning

In this section, we explore supervised network representation learning methods, i.e., graph neural networks (GNNs). The embeddings are optimized for the final task (e.g., career prediction). GNNs view the nodes of a specific type as labels, e.g., career nodes, and remove them from the network, while unsupervised network representation methods consider them as nodes. While unsupervised network representation methods consider career recommendation as link prediction, GNNs formulates it as node classification. To reduce bias, we directly incorporate a fairness objective in addition to an accuracy-based objective. In the following, we describe our method to make GNNs based embeddings more fair.

Graph Neural Networks Graph Neural Network (GNN) (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017b; Velickovic et al. 2018) is a powerful tool for graph mining. It has gained increasing popularity in various applications, including social network, knowledge graph, recommender system, and biomedical research.

Let $G = (V, E)$ denote a graph with node attributes X_v for $v \in V$. Given a set of nodes $\{v_1, ..., v_n\}$ and its labels $\{y_1, ..., y_n\}$, the task of graph supervised learning is to learn a representation vector h_v that helps predict the label of the node v, $\hat{y} = g(h_v)$. GNNs use the graph connectivity as well as node features to learn a representation vector (i.e., embedding) h_v for every node $v \in G$. GNNs use a neighborhood aggregation approach, where the representation of node v is iteratively updated by aggregating the representations of v's neighboring nodes. After k iterations of aggregation, v's representation captures the structural characteristics of its k-hop network neighborhood. Formally, the k-th layer of a GNN is:

$$h_v^k = Combine\big(h_v^{k-1}, Agg(h_v^{k-1}, h_u^{k-1})\big), \qquad (7)$$

where h_v is the representation of node v at the k-th iteration/layer, and u represents neighbors of v, and $Combine$ is a function such as an MLP network, and Agg is an aggregation function such as summation that aggregates neighbors of v. We initialize $h_v^0 = X_v$. The accuracy loss function of GNNs is:

$$\mathcal{L}_{acc} = -\sum_i^N \log P(\hat{y}_i = y_i) \ . \qquad (8)$$

Fairness Loss There are many fairness definitions. Here we only consider demographic parity (Dwork et al. 2012b) and equal opportunity (Hardt, Price, and Srebro 2016).

Given a classifier, N^i denotes the number of users in protected group g_i. N_k^i denotes the number of samples which are predicted to be k and are in protected groups g_i. $N_{k,k}^i$ denotes the number of samples which are predicted correctly and are in protected group g_i.

$$\mathit{diff}_{dp} = \sum_k \left| \frac{N_k^0}{N^0} - \frac{N_k^1}{N^1} \right| \qquad (9)$$

$$\mathit{diff}_{eo} = \sum_k \left| \frac{N_{k,k}^0}{N_k^0} - \frac{N_{k,k}^1}{N_k^1} \right| \qquad (10)$$

The above counting method cannot be used as a loss function, because the gradient cannot be back propagated. Hence we use the probabilistic prediction $p(\hat{y}|x)$ of the model to replace the hard count. Hence, we define the fairness-aware loss for GNN as follows,

$$\mathcal{L}_{dp} = \sum_k \Bigg(\frac{\sum_{x \in D: A=0} P(\hat{y} = k|x)}{N^0} - \frac{\sum_{x \in D: A=1} P(\hat{y} = k|x)}{N^1} \Bigg)^2, \qquad (11)$$

$$\mathcal{L}_{eo} = \sum_k \Bigg(\frac{\sum_{x \in D: A=0, y=k} P(\hat{y} = k|x)}{N_k^0} - \frac{\sum_{x \in D: A=1, y=k} P(\hat{y} = k|x)}{N_k^1} \Bigg)^2, \qquad (12)$$

where x denotes a sample, and D denotes the dataset. We compute two final loss functions in our GNN models. The first one is a demographic parity based fairness-aware loss function.

$$\mathcal{L}_{acc} + \lambda_{dp} * \mathcal{L}_{dp}, \qquad (13)$$

where λ_{dp} is a trade-off hyper-parameter. The second one is an equal opportunity based fairness-aware loss function.

$$\mathcal{L}_{acc} + \lambda_{eo} * \mathcal{L}_{dp}, \qquad (14)$$

where λ_{eo} is a trade-off hyper-parameter.

Experiments

In this section, we study the behavior of the proposed HNE debiasing methods and their applications in automated career counseling. We tested the systems' performance on two datasets: a Facebook dataset and a MovieLens dataset.

Dataset	Facebook	MovieLens
# careers	48	14
# items (FB pages or movies)	99,756	3,677
# users	7,069	4,920
# male users	2,721	3,558
# female users	4,332	1,362
avg users per career	62.81	93.47
avg users per item	14.04	222.40
avg items per user	198.15	166.21

Table 1: Statistics of Facebook and MovieLens datesets.

Datasets

The Facebook dataset used in the study was collected as a part of the myPersonality project (Kosinski et al. 2015). The data was collected with an explicit opt-in consent for reuse for research purposes. To protect privacy, the data was also anonymized.

The Facebook dataset contains rich information about a Facebook user such as his/her demographics (e.g., gender), the Facebook pages he/she likes, and his/her declared career concentrations (e.g., English, Computer Science, Psychology). Since Facebook "likes" contain rich information about a person's interests and preferences of a wide range of items/topics (e.g., books, musics, celebrities, brands and TV shows), they can be used to suggest possible career concentrations for a new user.

As the myPersonality dataset is no longer publicly available, to facilitate research reproduciblity, we employed a second dataset, the movieLens dataset, which is publically available. It contains similar information to the Facebook dataset such as the gender of a movie reviewer, the movies he/she likes and his/her occupation.

Table 1 shows the statistics of each dataset after we clean the data. The Facebook network created from the Facebook dataset consists of 7069 user nodes, $99,756$ item (a.k.a FB page) nodes, and 48 career nodes. If a user likes a page, a link is created between the user and the page. If a user declared a career concentration on Facebook, we create a link between the user and the career concentration.

The MovieLens-1M dataset originally contains $6,040$ users, $3,900$ movies and 19 careers. After we remove some careers that our system should not recommend such as "retired," "unemployed," and "K-12 student," the resulting MovieLens network consists of 4920 user nodes, 3677 movie nodes, and 14 career nodes. Similarly, we add a link between a user and a movie if a user rated a movie. If a user has declared an occupation, we also add a link between the user and the occupation.

In both datasets, each user only has one career. In the Facebook dataset, the gender of a few users is missing, while all users in MovieLens have gender information.

Since our datasets are not very large, to fully utilize our ground truth data, we employed nested cross validation for model training, testing and hyper-parameter tuning. Specifically, we first split the whole dataset into two parts, where 40% of the data were used for embeddings training, and the remaining 60% were used for career prediction. For the data reserved for career prediction, we split it further into three folds, with one of the folds for testing, and the other two folds for training and validation. We further split the training and validation data into four folds with one of the folds as a validation set, and the other three folds for training. The proportion of embeddings training, training, validation, and test was 4:3:1:2.

For GNN based methods, to facilitate result comparison, the test sets were the same as those used in the sampling-based methods. Since GNNs do not need any data to train HNE first, the data for embeddings training (i.e., 40% of the entire data) was added into the training and validation data, so the proportion of training, validation, and test was 7:1:2.

Experimental Setup

We have implemented a total of 8 different methods. Among them, two are traditional HIN embedding methods without gender debiasing (M2V and GNN), and the rest are fair HIN embedding methods. Among the six fair HIN embedding methods, one is a naive baseline (called balance data), which can be used as a pre-processing step. The rest of the methods employ a single or a combination of the debiasing methods we proposed. The details of each method is described below.

Balance data: For each career, we randomly remove users from the advantaged group so that we have an equal number of male and female users in the embedding training data. In this way, we deliberately create a balanced dataset to remove gender bias. Note that only the embeddings' training dataset is balanced. The other datasets remain the same to facilitate result comparison. After creating a balanced dataset, we use M2V to predict career choices.

M2V: we use the traditional meta-path generation algorithm in Eq 2 to generate paths, and learn user embeddings and career embeddings using metapath2vec algorithm (Dong, Chawla, and Swami 2017). ***M2V + fair sampling***: we use fair meta-path generation algorithm shown in Eq 3 to generate paths, and the rest of the processes are the same as M2V.

Projection: we calculate the bias direction using Eq. 5 and subtract the component of the embedding in the bias direction. All sampling-based methods can use the projection method as a post-processing step.

GNNs: we use the GraphSaint algorithm (Zeng et al. 2019), a state-of-art GNN algorithm to predict career. We have also tried GraphSAGE (Hamilton, Ying, and Leskovec 2017a). However, the results were much worse than GraphSaint. So in this paper, we only report the results from GraphSaint.

GNN-demographic-parity: we combine the demographic-parity based fairness loss with an accuracy loss (Eq. 13) to train the GraphSaint model (Zeng et al. 2019).

GNN-equal-opportunity: we combine the equal-opportunity based fairness loss with an accuracy loss (Eq. 14) to train the GraphSaint model (Zeng et al. 2019).

For the non-GNN based methods above, after we learn the HNE, we train a multilayer perceptron (MLP) to predict the career. The inputs to the MLP are the user embeddings and the career embeddings learned by M2v. Since the output of the softmax layer in MLP is a probability distribution and each candidate career has a probability of being chosen, we rank all career candidates based on its probability.

For all sampling-based methods, we used the same hyper-parameters listed below. The dimension of embeddings was 128; the size of negative samples was 5; the context window size was 5. For all the GNN-based methods, we used two convolutional layers. The dimension of the hidden layer is 128. The features of nodes are generated using the following procedure. Each Facebook page or movie is associated with a title/description. We average the word embeddings of all the words in the text as its features. For each user, we average the features of all the linked items (e.g., FB page or movie) as his/her features. For each career, we average the features of all the linked users as its features. The dimension of the features was 50.

Model Selection

Many of the methods used in our experiment have hyper-parameters that we can tune. For M2V, we tune the number of walks and the length of each walk. For M2V + fair sampling, we tune the number of walks, the length of a walk and the ratio r in (Eq. 3). The range of the number of walks and the length of each walk are $\{10, 20, \cdots, 190, 200\}$. For the projection method, the hyper-parameters are the same as M2V or M2V + fair sampling. For GNN-demographic-parity and GNN-equal-opportunity, the hyperparameters are λ_{dp} and λ_{eo} respectively.

We use Bayesian optimization to tune hyper-parameters. For M2V and GNN, the loss function defined in Bayesian optimization is the negative mean reciprocal rank (MRR). The Bayesian optimization algorithm aims to select a hyper-parameter combination with a low loss value.

For M2V+fair sampling, M2V+projection, and M2V + fair sampling + projection, the loss function defined in Bayesian optimization is (1) demographic parity $+$ equal opportunity if MRR is within 95% of the MRR of M2V; (2) demographic parity $+$ equal opportunity $-$ $100 *$ MRR, if MRR is out of 95% of the MRR of M2V. The motivation of the loss function design is that if the model has less than 5% MRR lose, we neglect the MRR loss and encourage the system to find hyper-parameters that optimize fairness. Otherwise, we consider both MRR and fairness loss when searching for the best model.

We used grid search to tune hyper-parameters for GNN-based methods, where $\lambda_{dp}, \lambda_{eo} \in \{10, 20, \cdots, 190, 200\}$ and the search step size was 10.

Evaluation Metrics

We use one prediction accuracy measure and two fairness measures to assess the performance of all the methods.

Accuracy Measure: we use Mean Reciprocal Rank (MRR) as the measure of prediction accuracy. It is a statistical measure for evaluating an ordered list of items. MRR is computed as the mean of the multiplicative inverse of the rank of the correct answers. MRR is frequently used in information retrieval and recommender systems to assess the system output. To compute MRR, let r_i denote the rank of the ground truth (i.e, user i's career choice), and N denote the number of samples. Then $MRR = \frac{1}{N}\sum_i^N 1/r_i$.

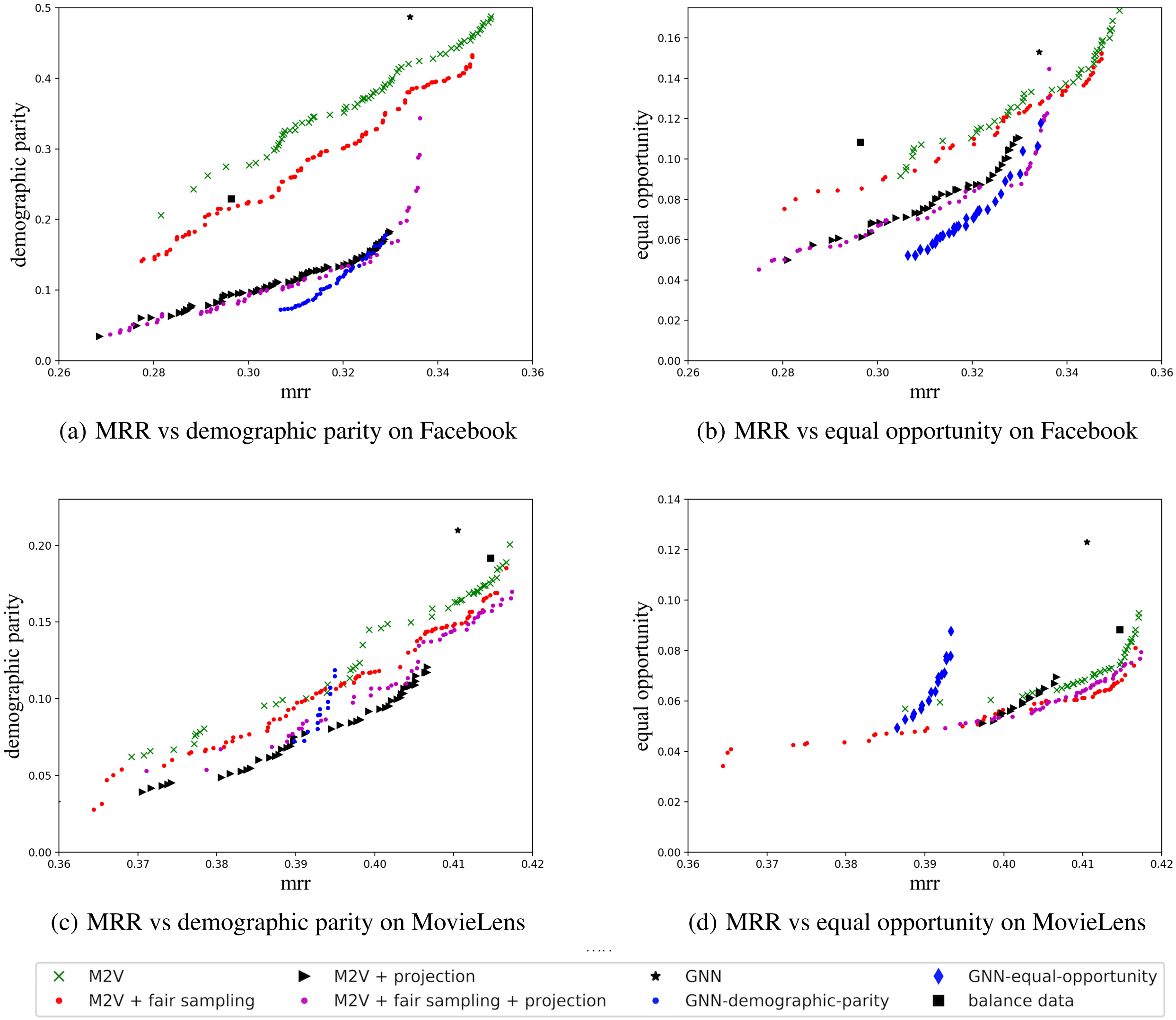
Fairness Measures: we use two widely used fairness measures in our evaluation: Demographic Parity and Equal Opportunity, which are defined in Eq. 9 and Eq. 10.

Experimental Results

To study the performance of different fair HNE algorithms, especially their ability to balance the tradeoff between prediction accuracy and the fairness, we plot the results on the test datasets in Figure 2(a)-2(d). The Y-axis of each chart represents a fairness measure (either demographic parity in Figure 2(a) and 2(c) or equal opportunity in Figure 2(b) and 2(d)), while the X-axis represents the prediction accuracy (MRR). Since some algorithms such as M2V and fair GNN approaches (e.g., GNN-Demograpic parity and GNN-Equal opportunity) have hyper-parameters that can be tuned to achieve different fairness and accuracy tradeoffs, we show the Pareto frontier of each method which consists of models that are not dominated by other alternatives from that method. We say that a model A dominates an alternative model B if model A outscores model B regardless of the trade-off between fairness and accuracy – that is, if A is better than B in both fairness and accuracy.

As shown in these charts, the traditional GNN method (represented as the *black star* in the charts) has relatively good MRR but poor fairness as it only tries to optimize prediction accuracy. The second baseline method that employs a naive data balancing technique to remove bias (represented as the *black square* in the charts) achieved moderate prediction accuracy as well as moderate fairness. The balance data model performed better on the MovieLens dataset than on the Facebook dataset because the Facebook dataset is more biased and thus a lot of data has to be removed from the advantaged group to achieve balance. Among the rest of the models, some methods such as the traditional sampling method (represented by the green crosses) and the fair sampling method (represented by red dots) are capable of achieving high MRR at the cost of low fairness (for both demographic parity and equal opportunity, the lower the value is, the more fair it is). Other methods, such as GNN-demographic parity/equal opportunity (represented by the blue dots/blue diamonds in the charts) and a combination of fair sampling and projection-based methods can achieve the best fairness and a reasonably good prediction accuracy.

To further illustrate each model's ability in balancing the tradeoff between accuracy and fairness and facilitate model comparison, we fixed the fairness dimension and only compare the prediction accuracy of these models. We choose three reference fairness thresholds to illustrate model performance under three conditions: high fairness (HF), medium fairness (MF) and low fairness (LF). The HF condition simulates a real world scenario where a system is used in making consequential decisions (e.g., sentencing). Thus under this condition, we may want to choose a model with high



(a) MRR vs demographic parity on Facebook

(b) MRR vs equal opportunity on Facebook

(c) MRR vs demographic parity on MovieLens

(d) MRR vs equal opportunity on MovieLens

Figure 2: The Pareto front of each method to demonstrate the accuracy and fairness tradeoff.

fairness. The low fairness condition simulates a real world scenario where the fairness of a system is not consequential (e.g., for entertainment).

The reference fairness thresholds were chosen based on the three baseline systems: *traditional GNN*, *balance data*, and *traditional sampling*. Since traditional GNN performed poorly on the fairness dimension, we use its fairness performance as the threshold to simulate the LF condition. In addition, since the balance data model normally achieves moderate fairness, we use its fairness performance as the threshold to simulate the MF condition. The high fairness (HF) threshold was obtained based on the traditional sampling method. We use the performance of its fairest model (the lowest *green crosses*) as the reference. Once we select the three fairness thresholds, we report the performance of the model with the highest MRR among all the models satisfying the fairness constraint.

As shown in Table 2, on the Facebook dataset, if under the LF condition, traditional methods without any debiasing such as M2V and GNN performed quite well. This is not surprising since the LF condition puts relatively little fairness constraints on the systems. Thus systems that totally ignore the fairness constraints (e.g., M2V and GNN) performed quite well. Under the HF condition, these two systems are among the worst performers. Some of the projection-based fair HNE methods such as M2V+projection, m2V+fair sampling+projection and GNN-based debiasing methods (e.g., GNN-demographic-parity) performed the best. Under the MF condition, again the projection-based debiasing method (e.g.,M2v+projection, M2V+fair sampling+ prjection) and the GNN-based debiasing methods (e.g., GNN-demographic parity) performed the best. In summary, the project-based

Method	dp_{LF}	dp_{MF}	dp_{HF}	eo_{LF}	eo_{MF}	eo_{HF}
balance data	0.2964	0.2964	–	0.2964	0.2964	0.2964
M2V	**0.3467**	0.2815	0.2815	0.3100	0.2885	–
M2V + fair sampling	**0.3364**	0.3003	0.3009	**0.3373**	0.3178	0.3003
M2V + projection	0.3222	0.3222	**0.3222**	0.3250	**0.3220**	**0.3016**
M2V + fair sampling + projection	0.3312	**0.3296**	**0.3260**	0.3273	0.3153	0.2982
GNN	**0.3341**	–	–	**0.3341**	–	–
GNN-demographic-parity	0.3262	**0.3262**	**0.3262**	0.3262	**0.3262**	**0.3262**
GNN-equal-opportunity	0.3300	**0.3254**	0.3190	**0.3300**	**0.3300**	**0.3300**

Table 2: Comparison of different HNE methods for career prediction using the Facebook network. Mean Reciprocal Rank (MRR) is reported under different demographic parity (dp)/equal opportunity (eo) constraints. Here, LF, MF, and HF represent the low fairness, medium fairness and high fairness conditions. Bold-faced numbers highlight the top 3 performers under each condition.

Method	dp_{LF}	dp_{MF}	dp_{HF}	eo_{LF}	eo_{MF}	eo_{HF}
balance data	**0.4147**	**0.4147**	–	**0.4147**	**0.4147**	–
M2V	**0.4125**	**0.4122**	0.3744	**0.4115**	**0.4125**	**0.4035**
M2V + fair sampling	0.4055	0.4055	**0.3946**	0.4055	0.4055	0.3971
M2V + projection	0.4049	0.4049	0.3891	0.4049	0.4049	**0.3988**
M2V + fair sampling + projection	0.4081	**0.4067**	**0.3908**	0.4081	**0.4081**	**0.4116**
GNN	**0.4105**	–	–	**0.4105**	–	–
GNN-demographic-parity	0.3916	0.3916	0.38963	0.3916	0.3916	0.38992
GNN-equal-opportunity	0.3921	0.3921	**0.39025**	0.3921	0.3911	0.39025

Table 3: Comparison of different HNE methods for career prediction using the MovieLens network. Mean Reciprocal Rank (MRR) is reported under different demographic parity (dp)/equal opportunity (eo) constraints. Here, LF, MF, and HF represent the low fairness, medium fairness and high fairness conditions. Bold-faced numbers highlight the top 3 performers under each condition.

and GNN-based debiasing methods performed quite well under both MF and HF conditions.

The results on the MovieLens dataset are slightly different. First, the baselines models such as M2V, balance data and GNN performed quite well under the LF condition. Both M2V and "balance data" also performed quite well under the MF condition. Under the HF condition, the projection-base method M2v+Fair sampling+projection consistently performed well regardless the fairness measures used.

As a summary, based on results on both the Facebook and the MovieLens dataset, GNN or M2V-based traditional embedding methods should be used if under the LF condition. Under the HF condition, project-based methods, such as M2V+projection or M2V+fair sampling +projection, consistently performed well. They also performed quite well under the MF condition.

Conclusion

Heterogeneous Network Embedding (HNE) is a popular technology that has been widely used in complex network mining. So far, little attention has been paid to the biases in HNE as well as its potential impact to downstream applications. Our research represents a first effort to address this issue. In this paper, we systematically investigated a wide range of HNE debiasing algorithms, ranging from sampling-based, projection-based, to graph neural network-based approaches. We evaluated the effectiveness of these methods in an automated career counseling task where we mitigate harmful gender bias in career recommendation. Based on the evaluation results on two datasets, we identified different algorithms that are effective under different conditions, which provides valuable guidance to practitioners.

References

Alshabani, N., and Soto, S. 2020. Early 20th-century career counseling for women: Contemporary practice and research implications. *Career Development Quarterly* 68(1):78 – 93.

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica, May* 23.

Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NeurIPS*, 4349–4357.

Buolamwini, J., and Gebru, T. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAT*, 77–91.

Calders, T., and Verwer, S. 2010. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21(2):277–292.

Caliskan, A.; Bryson, J. J.; and Narayanan, A. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356(6334):183–186.

Cavallari, S.; Zheng, V. W.; Cai, H.; Chang, K. C.-C.; and Cambria, E. 2017. Learning community embedding with community detection and node embedding on graphs. In *CIKM*, 377–386.

Dastin, J. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*.

Dev, S., and Phillips, J. 2019. Attenuating bias in word vectors. *arXiv preprint arXiv:1901.07656*.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 135–144. ACM.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012a. Fairness through awareness. In *ITCS*, 214–226. ACM.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012b. Fairness through awareness. In *ITCS*, ITCS '12, 214–226. Association for Computing Machinery.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *KDD*, 259–268. ACM.

Fu, T.-y.; Lee, W.-C.; and Lei, Z. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*, 1797–1806. ACM.

Gonen, H., and Goldberg, Y. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862* 609–614.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017a. Inductive representation learning on large graphs. In *NeurIPS*, 1024–1034.

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

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. In *NeurIPS*, 3315–3323.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks.

Kosinski, M.; Matz, S. C.; Gosling, S. D.; Popov, V.; and Stillwell, D. 2015. Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines. *American Psychologist* 70(6):543.

Liu, M.; Liu, S.; Zhu, X.; Liao, Q.; Wei, F.; and Pan, S. 2016. An uncertainty-aware approach for exploratory microblog retrieval. *IEEE Transactions on Visualization and Computer Graphics* 22(1):250–259.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 3111–3119.

Noble, S. 2018. *Algorithms of Oppression: How Search Engines Reinforce Racism*. NYU Press.

Papakyriakopoulos, O.; Hegelich, S.; Serrano, J. C. M.; and Marco, F. 2020. Bias in word embeddings. In *FAT*, 446–457.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD*, 701–710.

Rahman, T.; Surma, B.; Backes, M.; and Zhang, Y. 2019. Fairwalk: towards fair graph embedding. In *IJCAI*, 3289–3295.

Shi, Y.; Gui, H.; Zhu, Q.; Kaplan, L.; and Han, J. 2018. Aspem: Embedding learning by aspects in heterogeneous information networks. In *ICDM*, 144–152. SIAM.

Shi, C.; Hu, B.; Zhao, W. X.; and Yu, P. S. 2019. Heterogeneous information network embedding for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 31(2):357–370.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph attention networks. In *ICLR*.

Wang, H.; Zhang, F.; Hou, M.; Xie, X.; Guo, M.; and Liu, Q. 2018. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In *ICWSM*, 592–600.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. 2020. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Xie, Q.; Dai, Z.; Du, Y.; Hovy, E.; and Neubig, G. 2017. Controllable invariance through adversarial feature learning. In *NeurIPS*, 585–596.

Xiong, Y.; Guo, M.; Ruan, L.; Kong, X.; Tang, C.; Zhu, Y.; and Wang, W. 2019. Heterogeneous network embedding enabling accurate disease association predictions. *BMC Medical Genomics* 12(10):186.

Zafar, M. B.; Valera, I.; Gomez Rodriguez, M.; and Gummadi, K. P. 2017a. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*, 1171–1180.

Zafar, M. B.; Valera, I.; Rodriguez, M. G.; and Gummadi, K. P. 2017b. Fairness constraints: Mechanisms for fair classification. 962–970.

Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning fair representations. In *ICML*, 325–333.

Zeng, H.; Zhou, H.; Srivastava, A.; Kannan, R.; and Prasanna, V. 2019. Graphsaint: Graph sampling based inductive learning method. In *ICLR*.

Zhang, Y.; Xiong, Y.; Kong, X.; Li, S.; Mi, J.; and Zhu, Y. 2018. Deep collective classification in heterogeneous information networks. In *WWW*, 399–408.

Zhou, D.; Orshanskiy, S. A.; Zha, H.; and Giles, C. L. 2007. Co-ranking authors and documents in a heterogeneous network. In *ICDM*, 739–744.