Fairness in Graph Mining: A Survey

Yushun Dong, Jing Ma, Chen Chen, and Jundong Li

Abstract—Graph mining algorithms have been playing a significant role in myriad fields over the years. However, despite their promising performance on various graph analytical tasks, most of these algorithms lack fairness considerations. As a consequence, they could lead to discrimination towards certain populations when exploited in human-centered applications. Recently, algorithmic fairness has been extensively studied in graph-based applications. In contrast to algorithmic fairness on independent and identically distributed (i.i.d.) data, fairness in graph mining has exclusive backgrounds, taxonomies, and fulfilling techniques. In this survey, we provide a comprehensive and up-to-date introduction of existing literature under the context of fair graph mining. Specifically, we propose a novel taxonomy of fairness notions on graphs, which sheds light on their connections and differences. We further present an organized summary of existing techniques that promote fairness in graph mining. Finally, we summarize the widely used datasets in this emerging research field and provide insights on current research challenges and open questions, aiming at encouraging cross-breeding ideas and further advances.

Index Terms—Algorithmic Fairness, Graph Mining, Debiasing

arXiv:2204.09888v1 [cs.LG] 21 Apr 2022

1 Introduction

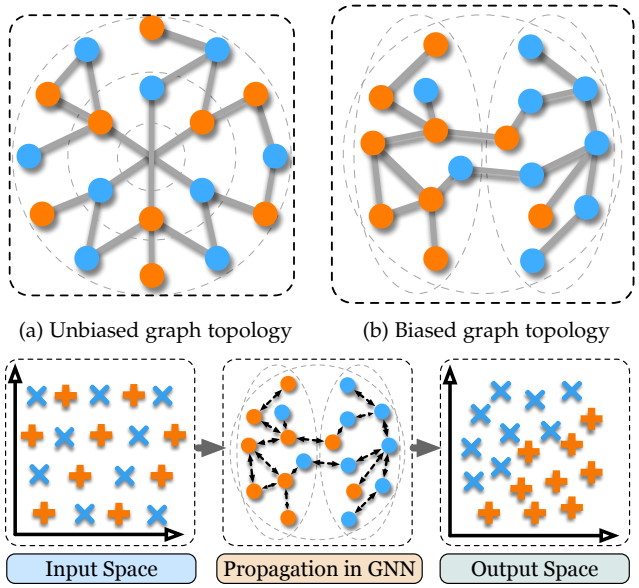
Graph-structured data is pervasive in diverse real-world applications, e.g., E-commerce [94], [112], health care [36], [51], traffic forecasting [66], [92], and drug discovery [15], [162]. In recent years, a number of graph mining algorithms have been proposed to gain a deeper understanding of such data. These algorithms have shown promising performance on various graph analytical tasks such as node classification [56], [79], [119], [152] and link prediction [4], [95], [96], [100], which contribute to great advances in many graph-based applications.

Despite the success of these graph mining algorithms, most of them lack fairness considerations. Consequently, they could yield discriminatory results towards certain populations when such algorithms are exploited in human-centered applications [74]. For example, a social network-based job recommender system may unfavorably recommend fewer job opportunities to individuals of a certain gender [89] or individuals in an underrepresented ethnic group [141]. With the widespread usage of graph mining algorithms, such potential discrimination could also exist in other high-stake applications such as disaster response [150], criminal justice [3], and loan approval [127]. In these applications, critical and life-changing decisions are often made for the individuals involved. Therefore, how to tackle unfairness issues in graph mining algorithms naturally becomes a crucial problem.

Fulfilling fairness in graph mining can be non-trivial due to two main challenges. The first challenge is to formulate proper fairness notions as the criteria to determine the existence of unfairness (i.e., bias). Although a vast amount of traditional algorithmic fairness notions have been proposed in the context of independent and identically distributed (i.i.d.) data [41], [102], they are unable to reflect the relational information (i.e., the topology) in graph data. For example, the same population can be connected with different topologies as in Fig. 1a and 1b, where each node represents an individual, and the color of nodes denotes their demographic subgroup membership, such as different genders. Compared with the graph topology in Fig. 1a, the topology in Fig. 1b has more intra-group edges than inter-group edges. The dominance of intra-group edges in the graph topology is a common type of bias existing in real-world graphs [38], [40], [65], which cannot be captured by traditional algorithmic fairness notions. The second challenge is to prevent the graph mining algorithms from inheriting the bias exhibited in the input graphs [40], [103], [139], [151]. We present a toy example to demonstrate how the information propagation mechanism in Graph Neural Networks (GNNs) [60], [79], [152] induces bias to the output node embeddings from a biased graph topology in Fig. 1c. In the input space, the node features are uniformly distributed. However, when the information propagation is performed on a biased topology as in Fig. 1b, the information received by nodes in different subgroups could be biased [40], leading to a biased embedding distribution in the output space.

There has been emerging research interest in fulfilling algorithmic fairness in graph mining. Nevertheless, the studied fairness notions vary across different works, which can be confusing and impede further progress. Meanwhile, different techniques are developed in achieving various fairness notions. Without a clear understanding of the corresponding mappings, future fair graph mining algorithm design can be difficult. Therefore, a systematic survey of recent advances is needed to shed light on future research. In this survey, we present a comprehensive and up-to-date review of existing works in fair graph mining. The main

- *Y. Dong is with Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, Virginia, US.*
  E-mail: yd6eb@virginia.edu
- *J. Ma is with Department of Computer Science, University of Virginia, Charlottesville, Virginia, US.*
  E-mail: jm3mr@virginia.edu
- *C. Chen is with Biocomplexity Institute, University of Virginia, Charlottesville, Virginia, US.*
  E-mail: zrh6du@virginia.edu
- *J. Li is with Department of Electrical and Computer Engineering, Department of Computer Science, and School of Data Science, University of Virginia, Charlottesville, Virginia, US.*
  E-mail: jundong@virginia.edu



(a) Unbiased graph topology (b) Biased graph topology

(c) An illustrative example of biased node embeddings (learned through information propagation mechanism of GNNs) induced by biased input graph.

Fig. 1: Examples of (a) unbiased graph topology, (b) biased graph topology, and (c) how information propagation mechanism induces bias in GNNs. Nodes in two different demographic subgroups are in orange and blue.

contributions of this survey paper are summarized as:

- ***Novel Fairness Taxonomy.*** We propose a novel taxonomy of fairness notions in graph mining. Such a taxonomy includes five groups of fairness notions: group fairness, individual fairness, counterfactual fairness, degree-related fairness, and application-specific fairness. For each group of fairness notions, we present their definitions and common quantitative metrics.
- ***Comprehensive Technique Review.*** We provide a comprehensive and organized review of six groups of techniques that are commonly utilized to promote fairness in graph mining algorithms. For each group of techniques, we summarize representative formulations under different fairness notions.
- ***Rich Public-Available Resources.*** We collect rich resources of benchmark datasets that can be employed for fair graph mining research. Therefore, this survey can facilitate the future development of new approaches to promote fairness in graph-based applications.
- ***Challenges and Future Directions.*** We present the limitations of current research and point out pressing challenges. Open research questions are also discussed for further advances.

Connections to Existing Surveys. Although there are several survey papers on algorithmic fairness [28], [35], [41], [102], [109], [120], they mainly focus on the fairness in the context of i.i.d. data. A few other surveys pay attention to the algorithmic fairness in relational data [121], [172]. Nevertheless, they are limited within either a certain application scenario (e.g., recommender systems) or a certain type of graph mining algorithm (e.g., machine learning-based algorithms). Different from them, our survey includes a detailed and systematic review of fairness in graph mining algorithms and corresponding applications.

TABLE 1: Notations and the corresponding definitions or descriptions.

Notations	Definitions or Descriptions
$\lvert\cdot\rvert$	Cardinality operator for any set.
$E[\cdot]$	Expectation operator.
$<\cdot,\cdot>$	Inner product operator.
$\mathcal{G}$	The graph data.
$\mathcal{V}$	The set of nodes.
$\mathcal{E}$	The set of edges.
$\mathcal{X}$	The set of node features.
$\mathcal{A}$	The seed set in influence maximization.
$\mathcal{N}_{v_i}$	The one-hop neighboring node set of v_i.
$\mathcal{V}_i$	The node set of the i-th sensitive subgroup.
$\mathbf{A}$	The adjacency matrix of graph $\mathcal{G}$.
$\mathbf{A}^\top$	The transpose of adjacency matrix.
$\mathbf{X}$	The node feature matrix of graph $\mathcal{G}$.
$\mathbf{z}_i$	The embedding of node v_i.
v_i	The i-th node.
n	The size of the node set $\mathcal{V}$.
d	The number of node features.
c	The class number for node classification.

Intended Audiences. The intended audiences of this survey are (1) researchers who would like to understand how fairness is defined and fulfilled in graph mining; and (2) practitioners who plan to generalize fair graph mining approaches to different applications.

Survey Structure. The remainder of this survey paper is organized as follows. Section 2 introduces the notations and preliminaries. In Section 3, different fairness notions and corresponding metrics are systematically reviewed. Based on these fairness notions and metrics, Section 4 introduces six groups of techniques to fulfill fairness. Rich open-source benchmark datasets for fairness studies in graph mining are summarized in Section 5. Section 6 discusses existing research challenges and open questions for future research. Finally, Section 7 presents the conclusion of this survey.

2 Notations and Preliminaries

In this section, we present important notations used throughout this paper and preliminaries. The notations and their definitions (or descriptions) are in Table 1.

Notations. We use bold uppercase letters (e.g., $\mathbf{A}$) and bold lowercase letters (e.g., $\mathbf{z}$) to denote matrices and vectors, respectively. For any matrix, e.g., $\mathbf{A}$, we refer to its (i,j)-th entry as $\mathbf{A}_{i,j}$, and the transpose of $\mathbf{A}$ as $\mathbf{A}^\top$. We use letters in calligraphy font (e.g., $\mathcal{V}$) to denote sets.

Preliminaries. In this paper, we use the terminology of "graph" and "network" interchangeably. We denote a plain graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ represent the set of nodes and edges, respectively. We use $\mathbf{A} \in \{0,1\}^{n\times n}$ to represent the graph adjacency matrix, where n is the total number of nodes, and $\mathbf{A}_{i,j} = 1$ implies that there exists an edge between node v_i and node v_j. For graph data with node features (i.e., attributed networks), we use a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ to denote the node feature matrix, where d is the number of node features. Unless otherwise specified, for the convenience of discussion on human-centered fairness, we consider that each individual is represented as a node

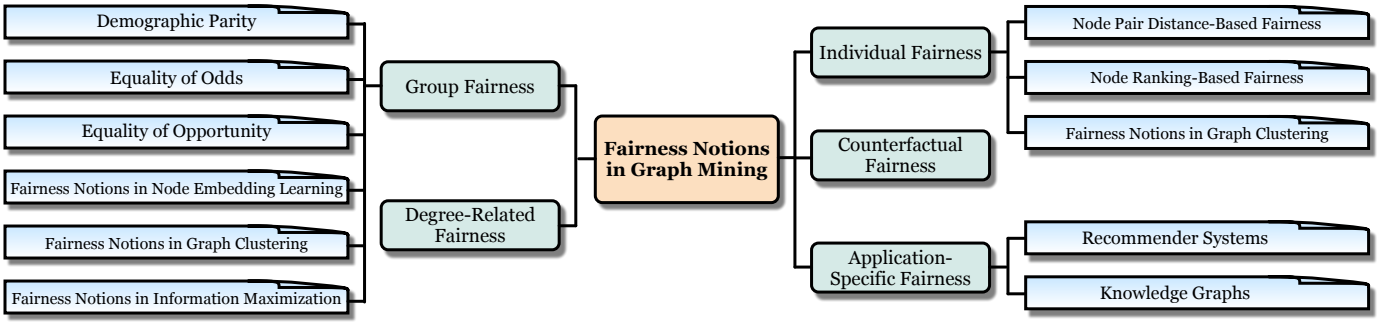


Fig. 2: Taxonomy of algorithmic fairness notions in graph mining algorithms.

in a graph by default. However, the literature in this survey and our discussion are not limited in this case. Additional clarifications will be presented in other scenarios.

3 Fairness Notions in Graph Mining

In this section, we propose a novel taxonomy that categorizes existing fairness notions in graph mining into five groups, as presented in Fig. 2. Following our proposed taxonomy, we hereby organize and introduce these fairness notions and their corresponding metrics.

3.1 Group Fairness

In many high-stake applications (e.g., loan approval systems [155], [163]), certain features (e.g., race and gender) are protected by law to avoid being abused [28], [41], [70], [102]. Additionally, in applications such as online social networking, there are a number of features that users are usually unwilling to share (e.g., occupation and age) [38]. These features are considered as protected or sensitive features [102]. Based on these features, the population can be divided into different demographic subgroups. Here we refer to these subgroups as sensitive subgroups. *Group Fairness* is then defined upon such sensitive subgroups. Generally speaking, group fairness requires that the algorithm should not yield discriminatory predictions or decisions against individuals from any specific sensitive subgroup [43]. In this section, we introduce popular fairness notions under group fairness.

3.1.1 Demographic Parity

Demographic Parity (a.k.a. *Statistical Parity* and *Independence*) is first introduced as a notion of group fairness based on binary sensitive feature(s) in binary classification tasks [43]. The binary sensitive feature divides the population into two sensitive subgroups (e.g., male/female). In binary classification tasks such as deciding whether a student should be admitted into a university or not, demographic parity is considered as achieved if the model yields the same acceptance rate for individuals in both sensitive subgroups. In graph mining, we first introduce demographic parity in node classification, followed by several extensions.

Demographic Parity in Node Classification. In node classification, we assume $\hat{Y}, S \in \{0, 1\}$ are the random variables representing the predicted class label and sensitive feature of a random node in the input graph, respectively. The criterion of demographic parity is then formulated as

$$P(\hat{Y} = 1|S = 0) = P(\hat{Y} = 1|S = 1). \tag{1}$$

To quantify to what extent the demographic parity is satisfied, Δ_{DP} is defined when both the predicted labels and sensitive feature(s) are binary [38], [88]. The formulation of Δ_{DP} is given as

$$\Delta_{DP} = |P(\hat{Y} = 1 \mid S = 0) - P(\hat{Y} = 1 \mid S = 1)|. \tag{2}$$

The intuition here is to measure the acceptance rate difference between the two sensitive subgroups. However, the applicable scenarios can be limited if only binary sensitive attributes are considered. In this regard, several following works extended demographic parity to multi-class sensitive feature scenarios [122], [136]. The rationale is that the acceptance rates given by the algorithm should be the same across all sensitive subgroups. To quantify demographic parity for multiple sensitive subgroups, Rahman et al. [122] leveraged the variance of acceptance rates across all sensitive subgroups, while Spinelli et al. [136] employed the largest acceptance rate difference among all subgroup pairs.

Extension to Link Prediction. In addition to the node classification task, demographic parity is also extended to the link prediction problem [87]. Specifically, we can obtain the average linking probability of node pairs spanning across different sensitive subgroups. Then demographic parity is achieved when such probability is the same for any two pairs of sensitive subgroups [25], [153]. Formally, assume (i, j) and (k, l) are the indices of two sensitive subgroup pairs (indices values can be the same within each tuple). The criterion of demographic parity is given as

$$\delta_{i,j} = \delta_{k,l}, \ \forall i, j, k, l, \tag{3}$$

where $\delta_{i,j}$ is the average linking probability of node pairs spanning across the i-th and the j-th sensitive subgroup, and it is formally defined as

$$\delta_{i,j} = \frac{1}{N_{i,j}} \sum_{v \in \mathcal{V}_i} \sum_{v' \in \mathcal{V}_j} P(f_{\text{link}}(v, v') = 1). \tag{4}$$

Here $P(f_{\text{link}}(v, v') = 1)$ is the probability that the edge (v, v') exists according to the link prediction model f_{link}; $\mathcal{V}_i$ and $\mathcal{V}_j$ represent the i-th and the j-th sensitive subgroup, respectively; $N_{i,j}$ is the number of node pairs spanning across $\mathcal{V}_i$ and $\mathcal{V}_j$, which is formally given as

$$N_{i,j} = |\{(v, v') \mid v \in \mathcal{V}_i, v' \in \mathcal{V}_j\}|. \tag{5}$$

We then introduce a metric $\Delta_{DP}^{\text{link}}$ that quantifies demographic parity in link prediction [25], [128], [153]. Specifi-

cally, $\Delta_{DP}^{\text{link}}$ is defined as the largest absolute difference [153] between all pairs of $\delta_{i,j}$ and $\delta_{k,l}$, which is defined as

$$\Delta_{DP}^{\text{link}} = \max_{\forall i,j,k,l} |\delta_{i,j} - \delta_{k,l}|. \quad (6)$$

Besides, a relaxed criterion of demographic parity has been defined by only focusing on intra- and inter-subgroup links [87]. Specifically, denote two random nodes as v and v', where $(v, v') \in \mathcal{V} \times \mathcal{V}$. Assume s and s' are the sensitive feature values of node v and v', respectively. A relaxed criterion for demographic parity is defined as

$$P(f_{\text{link}}(v, v') = 1 | s = s') = P(f_{\text{link}}(v, v') = 1 | s \neq s'). \quad (7)$$

Generally, such a criterion requires that for a random pair of nodes, the probability that they are connected should be the same, regardless of whether their sensitive feature values are the same or not. To quantify how well the criterion in Eq. (7) is satisfied, *Disparate Impact* (DI) and *Balanced Error Rate* (BER) are proposed in [87]. Specifically, DI quantifies to what level the link prediction model f_{link} prefers to give positive predictions for nodes with different sensitive feature values compared with those with the same ones. It is formally formulated as

$$\text{DI}\left(f_{\text{link}}, \mathcal{G}\right) = \frac{P(f_{\text{link}}(v, v') = 1 | s \neq s')}{P(f_{\text{link}}(v, v') = 1 | s = s')}. \quad (8)$$

A similar strategy is also adopted by BER [87], which measures the difference between $P(f_{\text{link}}(v, v') = 1 | s = s')$ and $P(f_{\text{link}}(v, v') = 1 | s \neq s')$.

Extension to Continuous Sensitive Feature(s). Although demographic parity has been widely used, most related studies are based on categorical (either binary or multi-class) sensitive feature(s). To extend the notion of demographic parity to continuous sensitive feature(s), Jiang et al. [68] proposed *Generalized Demographic Parity* (GDP) in node classification tasks. Specifically, for any sensitive feature value s, GDP requires that the difference between $E[\hat{Y} \mid S = s]$ and $E[\hat{Y}]$ should be as small as possible, where $E[\cdot]$ is the expectation operator. We then introduce ΔGDP to quantify how well GDP is achieved:

$$\Delta GDP = \int_0^1 \left| E[\hat{Y} \mid S = s] - E[\hat{Y}] \right| f_S(S = s) \mathrm{d}S. \quad (9)$$

The value of the sensitive feature S is assumed to be continuous and normalized between 0 and 1; $f_S(S = s)$ is the value of the sensitive feature PDF at $S = s$. Typically, a smaller value of ΔGDP indicates a higher level of GDP for the corresponding algorithm.

3.1.2 Equality of Odds

Equality of Odds is first introduced as a group fairness notion by Hardt et al. [61] in binary classification tasks. In general, the algorithm predictions are enforced to be independent with the sensitive feature(s) conditional on the ground truth class labels. The rationale is to prohibit the model from abusing the sensitive feature as a proxy of class labels for prediction. We introduce equality of odds in node classification tasks as follows.

Equality of Odds in Node Classification. In node classification, suppose that the predictions of the graph mining algorithm $\hat{Y}$, the ground truth labels Y, and the sensitive feature S are all binary. Equality of odds requires that

$$P(\hat{Y} = 1 | S = 0, Y = y) = P(\hat{Y} = 1 | S = 1, Y = y) \quad (10)$$

holds for both $y = 0$ and $y = 1$. In other words, Eq. (10) enforces predictions to bear equal TPR (i.e., True Positive Rate) and FPR (i.e., False Positive Rate) for the two sensitive subgroups. To quantify how well the equality of odds is satisfied, the largest difference of TPR (and FPR) between any two sensitive subgroups is often considered [88], [103]. The notion of equality of odds has also been extended to multi-class scenarios [136], where the value of TPR (and FPR) is required to be the same across all sensitive subgroups. To compute equality of odds in multi-class scenarios, we first calculate the maximum TPR difference and the maximum FPR difference between any two subgroups. Equality of odds is then measured by the larger difference value [136].

3.1.3 Equality of Opportunity

Equality of Opportunity extends the notion of equality of odds [61]. Specifically, in binary classification tasks, equality of opportunity only requires the positive predictions to be independent of sensitive feature(s) for individuals with positive ground truth labels [61]. We introduce equality of opportunity in node classification tasks as follows.

Equality of Opportunity in Node Classification. The criterion of equality of opportunity is given as

$$P(\hat{Y} = 1 | S = 0, Y = 1) = P(\hat{Y} = 1 | S = 1, Y = 1). \quad (11)$$

It should be noted that in most cases, $\hat{Y} = 1$ is an advantaged prediction [61]. Therefore, the intuition of equality of opportunity can be interpreted as: we want to avoid assigning disadvantaged predictions to individuals qualified for advantaged ones only because of their sensitive subgroup membership. In this regard, equality of opportunity is often advocated for economic justice [12], and a typical application scenario is job candidate selection. We then introduce a commonly employed quantitative metric Δ_{EO} for equality of opportunity in node classification. Specifically, Δ_{EO} measures how far the prediction deviates from the ideal situation that satisfies equality of opportunity. Δ_{EO} is formally given as

$$\begin{aligned} \Delta_{EO} = |&P(\hat{Y} = 1 \mid Y = 1, S = 0) \\ &- P(\hat{Y} = 1 \mid Y = 1, S = 1)|. \end{aligned} \quad (12)$$

Extension to Link Prediction. Equality of opportunity has been extended to link prediction tasks [25]. At a high level, equality of opportunity in link prediction is achieved when the TPR of link prediction is independent of the underlying sensitive feature. Formally, the node pairs with predicted positive links are regarded as the instances with $\hat{Y} = 1$, while the node pairs connected with actual links are the instances with $Y = 1$. Denote $N_{i,j} = |\{(v_l, v_m)|, v_l \in \mathcal{V}_i, v_m \in \mathcal{V}_j, \mathbf{A}_{l,m} = 1\}|$, which is the number of ground truth edges between the i-th and the j-th sensitive subgroups. For node pairs spanning across the i-th and the j-th subgroups, the TPR is given as

$$\epsilon_{i,j} = \frac{1}{N_{i,j}} \sum_{v_l \in \mathcal{V}_i} \sum_{v_m \in \mathcal{V}_j} P_{v_l, v_m} \mathbb{1}(\mathbf{A}_{\alpha,\beta} = 1). \quad (13)$$

Here, $\mathbb{1}(\cdot)$ is the identity function; P_{v_l,v_m} is the predicted probability that node v_l and v_m are linked. Generally, if the TPRs of any two sensitive subgroup pairs are the same, then they are independent of the sensitive subgroup membership. Denote $(i,\ j)$ and $(k,\ l)$ as the indices of two sensitive subgroup pairs (indices values can be the same within each tuple). The criterion of equality of opportunity is then formulated as

$$\epsilon_{i,j} = \epsilon_{k,l},\ \forall i,j,k,l. \tag{14}$$

To quantitatively measure the equality of opportunity in link prediction, Woodworth et al. [153] proposed $\Delta_{EO}^{\text{link}}$, which is formally defined as

$$\Delta_{EO}^{\text{link}} = \max_{\forall i,j,k,l} |\epsilon_{i,j} - \epsilon_{k,l}|. \tag{15}$$

3.1.4 Group Fairness in Node Embedding Learning

Learning fair node embeddings has received much research attention in recent years, as these fair embeddings can be employed for various downstream tasks to achieve fair results. Nevertheless, as node embeddings are mostly leaned without using the node label information, traditional fairness notions such as demographic parity (introduced in Section 3.1.1), equality of odds (introduced in Section 3.1.2), and equality of opportunity (introduced in Section 3.1.3) cannot be directly grafted. Here we introduce two types of group fairness notions for node embeddings.

Distribution-Based Fairness. A common criterion of fulfilling fairness for node embeddings is that the distributions of learned embeddings from different sensitive subgroups are close to each other [40], [42], [44], [110]. Generally, if the distributions of embeddings from different sensitive subgroups are similar, these embeddings are then regarded as decoupled from the sensitive feature, i.e., they are fair in terms of group fairness. Typically, distribution-based fairness is measured by the difference between these embedding distributions, and a common metric to quantify distribution difference is Wasserstein distance [40], [44].

Model-Based Fairness. The intuition of model-based fairness is to train a new model (e.g., MLP classifier [18] or SVM [25]) to predict the sensitive feature value based on the obtained node embeddings. For such a new model, the incapability of the sensitive feature prediction (e.g., low prediction accuracy) indicates the decoupling between the node embeddings and the sensitive feature, which thus implies a high level of group fairness [159].

3.1.5 Group Fairness in Graph Clustering

Group fairness in graph clustering requires all sensitive subgroups to be proportionally represented by the nodes in each cluster [34], [80], [129]. In this scenario, a common fairness metric is *Balance Score* [80]. Formally, the balance score of the cluster $\mathcal{C}_k$ ($1 \leq k \leq K$, where K is the total cluster number) is given as

$$\text{balance}(\mathcal{C}_k) = \min_{i \neq i', i, i' \in \{1,\dots,H\}} \frac{|\mathcal{V}_i \cap \mathcal{C}_k|}{|\mathcal{V}_{i'} \cap \mathcal{C}_k|}, \tag{16}$$

where $\mathcal{V}_i$ denotes the node set of the i-th sensitive subgroup; H is the number of sensitive subgroups. Generally, the balance score of cluster $\mathcal{C}_k$ reflects the largest discrepancy of node numbers between any two sensitive subgroups in this cluster. Considering that $\min_k \text{balance}\,(C_k) \leq \min_{i \neq i'} |V_i| \,/\, |V_{i'}|$, a larger minimum balance score over all clusters indicates a higher level of fairness in clustering [34].

3.1.6 Group Fairness in Influence Maximization

Influence maximization algorithms have been adopted in various high-stake scenarios such as HIV prevention [166], financial inclusion [11], and disease transmission [10]. Recently, algorithmic fairness in influence maximization has also attracted much research attention. In influence maximization, a set of nodes are initiated as seeds in a graph, and each seed influences its neighboring nodes by a certain probability. Given a budget for the number of seed nodes, the goal of influence maximization is to find a seed node set to influence the largest number of nodes. Specifically, the notion of fairness in influence maximization can be defined from different perspectives as below.

Maxmin Fairness. *Maxmin Fairness* is first introduced by Tsang et al. [148] based on Rawlsian theory of justice [125]. Given the seed node set $\mathcal{A}$, the lowest ratio of the influenced nodes among all sensitive subgroups is given as

$$U_{\min} = \min_i \frac{\mathcal{I}_{\mathcal{G},\mathcal{V}_i}(\mathcal{A})}{|\mathcal{V}_i|}, \text{where } i \in \{1, ..., H\}. \tag{17}$$

Here H is the total number of sensitive subgroups; $\mathcal{I}_{\mathcal{G},\mathcal{V}_i}(\mathcal{A})$ is the expected number of the influenced nodes in the i-th sensitive subgroup based on $\mathcal{A}$. Maxmin fairness requires that $U_{\min}$ should be as large as possible. Meanwhile, the value of $U_{\min}$ is employed to quantify the level of maxmin fairness in [148]. However, it should be noted that achieving a large $U_{\min}$ can greatly jeopardize the goal of maximizing the influence over the whole population. For instance, Tsang et al. [148] pointed out that one sensitive subgroup can be poorly connected with other nodes in the graph. In such a case, many seed nodes would be assigned to this subgroup to promote the influence rate within such subgroup, despite the fact that they can be reassigned to other subgroups to achieve a higher influence rate over the whole population.

Diversity. Tsang et al. [148] proposed another fairness criterion named *Diversity Constraint*. Here, it is assumed that a budget of seed nodes is provided for each sensitive subgroup, and the budget size is proportional to the subgroup size. Denote the seed node set for $\mathcal{V}_i$ as $\mathcal{A}'_i$, where $|\mathcal{A}'_i| = \lceil K\,|\mathcal{V}_i|\,/|\mathcal{V}| \rceil$. Here K is the budget for the seed node set of the whole population (i.e., $\mathcal{A}$), and $\lceil \cdot \rceil$ is the ceiling function. Then the authors defined $\mathcal{I}_{\mathcal{G}[\mathcal{V}_i]}(\mathcal{A}'_i)$ as the expected number of the influenced nodes in $\mathcal{V}_i$, where these nodes can only be influenced by the seed nodes in $\mathcal{A}'_i$ via intra-group edges. Diversity constraint requires that the choice of $\mathcal{A}$ should satisfy $\mathcal{I}_{\mathcal{G},\mathcal{V}_i}(\mathcal{A}) \geq \max_{\mathcal{A}'_i} \mathcal{I}_{\mathcal{G}[\mathcal{V}_i]}(\mathcal{A}'_i)$ for all i. In other words, for each sensitive subgroup, $\mathcal{A}$ should achieve an influence rate larger or equal to the influence rate when this group is assigned a proportional number of seed nodes, given that the influence only flows via intra-group edges. If such criterion is satisfied, the sensitive subgroups with fewer (than a proportional number) seed nodes would still obtain enough amount of influence from other sensitive subgroups. The percentage of sensitive subgroups that

violates such criterion is used to measure the diversity as in [148].

Utility Difference-Based Fairness. A different desideratum of fair influence maximization is to ensure that the influenced node ratios in different sensitive subgroups are similar. *Demographic Parity in Influence Maximization* (DP) is defined following such idea [5], [123], [138], and its criterion is formulated as

$$|u_i(\mathcal{A}) - u_j(\mathcal{A})| \leq \delta,\ \forall i, j \in \{1, ..., H\},\ \text{where } \delta \in [0, 1). \tag{18}$$

Here $\mathcal{A}$ is the seed set within budget K. δ is the largest tolerance for the influenced node ratio difference between any two sensitive subgroups, while $u_i(\mathcal{A})$ is the expected ratio of influenced nodes in the i-th sensitive subgroup under seed set $\mathcal{A}$. To measure utility difference between different subgroups, *Maximum Disparity in Normalized Utilities* is proposed in [5]. Formally, it is defined as

$$\max_{i,j \in \{1,...,H\}} |u_i(\mathcal{A}) - u_j(\mathcal{A})|. \tag{19}$$

Generally, Eq. (19) measures the maximum disparity of the influenced node ratio between any two sensitive subgroups.

Seed Set-Based Fairness. In influence maximization, the selection of seed node set could also encounter unfairness issues. For example, if the propagation of HIV awareness is originated from a set of seed nodes with unbalanced sensitive subgroup membership, some sensitive subgroups could have the privilege to know critical knowledge about HIV much earlier than other sensitive subgroups. This indicates that a biased seed node set could potentially put some sensitive subgroups in disadvantaged situations [137]. Therefore, it is critical to study the fairness of seed node selection. A common criterion of seed set-based fairness is that the number of seed nodes in any sensitive subgroup should be proportional to its population [45], [137]. Formally, for $1 \leq i, j \leq H$ (H is the total number of sensitive subgroups), the criterion is given as

$$\frac{|\{v \in \mathcal{A} \mid v \in \mathcal{V}_i\}|}{|\mathcal{V}_i|} = \frac{|\{v \in \mathcal{A} \mid v \in \mathcal{V}_j\}|}{|\mathcal{V}_j|}. \tag{20}$$

To measure seed set-based fairness for sensitive subgroup $\mathcal{V}_i$, Stoica et al. [137] employed the discrepancy between $|\mathcal{A} \cap \mathcal{V}_i|/|\mathcal{V}_i|$ and $|\mathcal{V}_i|/|\mathcal{V}|$. Here a smaller discrepancy indicates a higher level of seed set-based fairness for $\mathcal{V}_i$.

3.2 Individual Fairness

Compared with group fairness, *Individual Fairness* does not consider any sensitive features. Instead, it focuses on fairness at the individual (e.g., each node in graph data) level [43]. Generally, individual fairness requires that similar individuals should be treated similarly. Hence, individual fairness is considered as a fairness notion at a finer granularity than group fairness. Currently, there are only a few graph mining algorithms that consider individual fairness. We hereby present several existing individual fairness notions.

3.2.1 Node Pair Distance-Based Fairness

A widely adopted definition of individual fairness is that the pair-wise node distances in the input space and output space should satisfy *Lipschitz Condition* [43], [73]. Specifically, Lipschitz condition requires that the distance of any node pairs in the output space should be smaller or equal to their corresponding distance in the input space (usually re-scaled by a scalar). Formally, given a pair of nodes v_i and v_j, Lipschitz condition is given as

$$D(f(v_i), f(v_j)) \leq L \cdot d(v_i, v_j), \tag{21}$$

where $f(\cdot)$ is the predictive model that gives the node level output (e.g., node embeddings). $D(\cdot, \cdot)$ and $d(\cdot, \cdot)$ are the distance metrics in the output and input space, respectively. L is the Lipschitz constant that re-scales the input distance between node v_i and v_j. In graph mining, Lipschitz condition formulates the guiding principle of individual fairness (i.e., treating similar input nodes similarly) by restricting the pair-wise output distance of nodes.

To measure individual fairness based on Lipschitz condition, Zemel et al. [170] first proposed *Consistency* on non-graph data. The intuition is to measure the average distance of the output between each individual and its k-nearest neighbors. Generally, for the algorithm outputs, a larger average distance indicates a lower level of individual fairness. In [170], consistency is defined as

$$1 - \frac{1}{n \cdot k} \sum_{i=1}^{n} \left| \hat{y}_i - \sum_{j \in k\text{NN}(\mathbf{x}_i)} \hat{y}_j \right|, \tag{22}$$

where $\hat{y}_i$ is the probabilistic classification output for node v_i; function $k\text{NN}(\cdot)$ takes the features of node v_i as input and returns the index set of its k-nearest neighbors in the feature space. In graph mining algorithms, a similar formulation of consistency is proposed by Lahoti et al. [88] based on similarity matrix $\mathbf{S}$. Generally, $\mathbf{S}$ describes the similarity between nodes in the input space and can be given based on attributes, graph topology, or knowledge from domain experts [73], [88]. Here, consistency is formulated given as

$$1 - \frac{\sum_i \sum_j |\hat{y}_i - \hat{y}_j| \cdot \mathbf{S}_{ij}}{\sum_i \sum_j \mathbf{S}_{ij}} \quad (i \neq j) \tag{23}$$

in binary node classification tasks. For any graph mining algorithm, a large value of consistency indicates that it gives similar outputs for similar nodes in the input space, i.e., the algorithm performs well on individual fairness. Apart from that, Kang et al. [73] proposed to use the similarity-weighted output discrepancy between nodes to measure unfairness, which is formulated as $\text{Tr}(\hat{\mathbf{Y}}^\top \mathbf{L}_\mathbf{S} \hat{\mathbf{Y}})$. Here $\hat{\mathbf{Y}}$ is the output matrix of the graph mining algorithm. Each row in $\hat{\mathbf{Y}}$ represents the output vector for the corresponding node. $\text{Tr}(\cdot)$ is the trace operator for any matrix. $\mathbf{L}_\mathbf{S}$ denotes the Laplacian matrix of the similarity matrix $\mathbf{S}$. Such a metric measures the weighted sum of pair-wise node distance in the output space, where the weighting score is the pair-wise node similarity. Hence for any graph mining algorithms, a smaller value of the similarity-weighted discrepancy typically implies a higher level of individual fairness.

3.2.2 Node Ranking-Based Fairness

Although Lipschitz condition (introduced in Section 3.2.1) has been widely used as the individual fairness criterion, there could be problematic in practical scenarios. Specifically, determining whether the outputs of two individuals

are similar or not based on absolute input distances can be inappropriate, as such criterion cannot calibrate across different individuals [39]. Besides, Lipschitz condition imposes the comparison between the distances in two different spaces, which makes the Lipschitz constant hard to be determined. To handle these drawbacks, Dong et al. [39] proposed to fulfill individual fairness from a ranking perspective. Specifically, a similarity matrix $\mathbf{S}$ is provided to describe the pair-wise similarity for individuals in the input space. Based on $\mathbf{S}$, each individual has a ranking list R_1 that indicates the relative similarity ranking between itself and others. Similarly, a corresponding ranking list R_2 can also be derived based on the pair-wise individual similarity in the output space. From the perspective of node ranking, individual fairness is regarded as fulfilled when the two ranking lists (R_1 and R_2) are the same for each individual [39]. Nevertheless, such a criterion is hard to be satisfied. In practice, the average top-k similarity between R_1 and R_2 over all individuals is adopted to measure individual fairness, where NDCG@k [64] and ERR@k [29] are two common ranking similarity metrics.

3.2.3 Individual Fairness in Graph Clustering

In clustering, the criterion of individual fairness is defined in a different way compared with other tasks. Specifically, for each node in a graph, if its neighbors are proportionally distributed to each cluster, individual fairness is then fulfilled [59]. Formally, under individual fairness, a clustering algorithm satisfies fair clustering for node v_i if

$$\frac{|\{v_j : \mathbf{A}_{i,j} = 1 \wedge v_j \in \mathcal{C}_k\}|}{|\mathcal{C}_k|} = \frac{|\{v_j : \mathbf{A}_{i,j} = 1\}|}{|\mathcal{V}|} \tag{24}$$

for all $k \in \{1, ..., K\}$. Here K represents the total number of clusters, and $\mathcal{C}_k$ represents the set of nodes in cluster k. The intuition here is that for each node, the ratio occupied by its one-hop neighbors in each cluster should be the same as the ratio occupied by its one-hop neighbors in the whole population. We then introduce the metric to quantify fair clustering. Specifically, ρ_i measures how disproportionately the one-hop neighbors of node v_i are assigned in different clusters [59]. Formally, ρ_i is defined as

$$\rho_i = \min_{k,l \in \{1,...,K\}} \frac{|\mathcal{C}_k \cap \mathcal{N}_{v_i}|}{|\mathcal{C}_l \cap \mathcal{N}_{v_i}|}, \tag{25}$$

where $\mathcal{N}_{v_i}$ denotes the neighboring node set of v_i. In [59], the average ρ_i across all nodes in the graph is employed as the metric of individual fairness in graph clustering.

3.3 Counterfactual Fairness

Different from the above fairness notions, *Counterfactual Fairness* [85] defines fairness from the causal perspective [118]. Specifically, counterfactual fairness is considered to be achieved when the prediction results for each individual and his/her counterfactuals ("counterfactuals" in this setting are different versions of the same individual when his/her sensitive feature had been changed to different values) are maintained to be the same. For example, the algorithmic decision for an applicant's loan application should be the same regardless of his/her race. Counterfactual fairness is defined based on Pearl's structural causal model [118], where a causal model describes the causal relations between different variables. For any variables A and B, the counterfactual "what would B have been if A had been set to a specific value a" is denoted by $B_{A \leftarrow a}$. Denote $\hat{Y}_{S \leftarrow s} = f(X_{S \leftarrow s}, s)$ as the model prediction made for the counterfactual when the sensitive feature S had been set to a s. Counterfactual fairness is defined as

$$P(\hat{Y}_{S \leftarrow s} = y | X = \mathbf{x}, S = s) = P(\hat{Y}_{S \leftarrow s'} = y | X = \mathbf{x}, S = s) \tag{26}$$

for all specific values $y, \mathbf{x}$ and $s' \neq s$.

Recently, there has been a line of works [3], [104], [173] that extend counterfactual fairness from traditional i.i.d. data to graph data. Most of these works aim to learn counterfactually fair node embeddings, and then make predictions based on the embeddings. Agarwal et al. [3] defined that a graph mining algorithm satisfies counterfactual fairness if the corresponding embedding for each node remains the same regardless of its sensitive feature(s) values (other features and graph structure stay unchanged). To step forward, Ma et al. [104] further considered more subtle issues that may cause counterfactual unfairness in graphs: (1) biases can be induced by each node's neighboring nodes in graphs; (2) biases can be induced by the causal relations between the sensitive feature(s) to other features or graph structure. Accordingly, this work defines graph counterfactual fairness for node embedding learning as follows. Given a graph encoder $\Phi(\cdot) : \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times d'}$, it satisfies graph counterfactual fairness if we have

$$\Phi(\mathbf{X}_{\mathbf{S} \leftarrow \mathbf{s}'}, \mathbf{A}_{\mathbf{S} \leftarrow \mathbf{s}'})_i = \Phi(\mathbf{X}_{\mathbf{S} \leftarrow \mathbf{s}''}, \mathbf{A}_{\mathbf{S} \leftarrow \mathbf{s}''})_i, \tag{27}$$

for any node v_i, where n is the number of nodes, $\mathbf{s}'$ and $\mathbf{s}''$ are arbitrary sensitive feature values of all nodes, where $\mathbf{s}', \mathbf{s}'' \in \{0, 1\}^n$ and $\mathbf{s}' \neq \mathbf{s}''$, $\Phi(\cdot)_i$ denotes the embedding of node v_i. Such a criterion requires the embeddings learned from the original graph and counterfactuals (where the sensitive feature values of any subset of the n nodes had been changed) to be the same.

To measure counterfactual fairness on graphs, recent works [3], [173] usually adopt *Unfairness Score*, which is the percentage of nodes whose predicted label changes when their sensitive feature values are changed (while other features are fixed). Beyond that, Ma et al. [104] also proposed to evaluate graph counterfactual fairness by measuring the average prediction discrepancy between any two different versions of counterfactual sensitive feature assignment on all the n nodes. However, as there are too many combinations for the sensitive feature values of all the nodes, and the true causal model is hardly available in the real world, such metric is commonly computed by approximation [104].

3.4 Degree-Related Fairness

Different from other traditional fairness notions, the study on *Degree-Related Fairness* is fairly new in the graph mining community. In networked data, if two nodes are connected, there could be dependency between them. Such dependency can be informative, and thus extracting the dependency between connected nodes benefits various tasks in graph mining. However, for low-degree nodes, their connections only contribute limited information on the dependency

between themselves and other nodes. In this regard, it can be difficult for graph mining algorithms to effectively capture critical information about these nodes, which often leads to worse utility compared with high-degree nodes. For example, the performance of GNNs in graph analytical tasks (e.g., node classification) on high-degree nodes (e.g., a celebrity who has a lot of followers) often deviates from that on low-degree nodes (e.g., an average Joe who has few followers) [31], [75], [145]. Correspondingly, degree-related fairness requires that nodes should bear similar utility (e.g., node classification accuracy) in the graph mining algorithms regardless of their degrees. To measure degree-related fairness, Kang et al. [75] leveraged the variance of average cross-entropy loss w.r.t. node degrees, and such metric is defined for node classification tasks.

3.5 Application-Specific Fairness

Apart from the application-agnostic fairness notions mentioned above, there are also other fairness notions particularly designed for certain graph-based applications. Here, we introduce application-specific fairness notions in recommender systems and knowledge graphs.

3.5.1 Fairness Notions in Recommender Systems

User Fairness. *User Fairness* is an indispensable fairness notion in recommender systems. Generally, user fairness requires that the recommendation quality for different users [52], [86], [93] should be similar. For example, in a recommender system, we can divide users into two groups according to their activity levels: active users and inactive ones. Usually, inactive users tend to receive unsatisfying recommendations compared with the active ones [52], [93], as they reveal less information about their preferences. Correspondingly, user fairness is usually measured by the recommendation quality discrepancy between active and inactive users [93]. Additionally, users can be divided into different sensitive subgroups according to their sensitive features. Yao et al. [167] proposed to measure the disadvantage level of each sensitive subgroup with the average deviation between the predicted item ratings and ground truth ratings among group members. The average disadvantage level across all subgroups is then employed to quantify the user unfairness. Furthermore, user fairness has also been studied in *Group Recommendation*. In general, the goal of group recommendation is to provide recommendations that comply with the preferences of most users in the group [6]. However, when the preferences of certain users (in a group) are ignored by the recommender system, they would feel that they are being unfairly treated. User fairness then requires that the preferences of each user in a given group should not be neglected by the recommendation algorithm. To measure user fairness for a group of users, Malecek et al. [105] leveraged the difference between the minimal and maximal user-item relevance scores across all users. Similar metrics are also adopted by other works such as [76], [97].

Popularity Fairness. *Popularity Fairness*, which requires that popular items should not be over-emphasized compared with other items, is another common fairness notion in recommender systems [33], [84], [157]. A well-known example of popularity unfairness issue is the *Filter Bubble* [116] problem, which describes the scenarios that users are isolated from less popular items or information [2], [101]. Typically, the level of popularity fairness is measured by the average recommendation rate of less popular instances (e.g., items, users, and social media posts) [1], [17], [54]. For example, in social platforms, Masrour et al. [101] extended the modularity score [111] on graph data to measure popularity fairness in friend recommendation. Formally, assume that a certain partition divides users into different groups, and $M(v_i)$ and $M(v_j)$ represent the group membership for user v_i and v_j, respectively. To measure how less popular user groups are connected with others, the metric of popularity fairness is formulated as

$$Q_{\text{fairness}} = \frac{1}{2|\mathcal{E}|}\sum_{i,j}(\mathbf{A}_{i,j} - \frac{d_i d_j}{2|\mathcal{E}|})\delta(M(v_i), M(v_j)), \quad (28)$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta function [147]; d_i and d_j indicate the degree of node i and j; $|\mathcal{E}|$ is the total number of edges in the graph data. Given a friend recommendation algorithm, a lower value of Q_{fairness} indicates that the algorithm yields more inter-group edges for the users in the graph. This implies that some less popular groups are encouraged to connect more with other groups, which relieves the filter bubble effect in friendship recommendation.

Provider Fairness. *Provider Fairness*, a.k.a. *Producer Fairness*, requires that items from different providers should receive the same exposure rate to the customers. Various strategies have been proposed to measure provider fairness. Patro et al. [117] set a minimum exposure guarantee for all providers and used the number of unsatisfied providers to measure provider fairness. Liu et al. [99] measured the provider diversity with the average number of providers appearing in recommendations. Boratto et al. [16] adopted both the user-item relevance difference and item exposure rate difference between different providers as the corresponding metrics.

Marketing Fairness. *Marketing Fairness* is another fairness notion in recommender systems proposed by Wan et al. [154]. In their paper, they pointed out that online shopping platform users are less likely to interact with items whose marketing strategy is not consistent with their identity. For example, some gender-neutral items (e.g., armband) could be marketed using images of males [57], [154]. Therefore, even if both male and female users are potential customers, female users tend to interact less with these items. In this context, male and female users are regarded as identity-consistent and identity-inconsistent users to the marketing content, respectively. Recommender systems could then inherit such bias from data and yield biased recommendations for female (i.e., identity-inconsistent) users in the future. To measure marketing fairness, Wan et al. [154] proposed to calculate the variance of recommendation errors for identity-consistent and identity-inconsistent users, and the variance discrepancy is adopted as the corresponding metric.

3.5.2 Fairness Notions in Knowledge Graphs

Social Fairness. Knowledge graph embeddings could encode historical *Social Biases* [8], [48] and one typical example is the stereotype that bankers are males and nurses are females [49]. Such biases have been observed in different knowledge graph-based tasks including entity embedding learning [48], [49] and word embedding learning based on knowledge graphs [14], [26], [53]. In recent years, various

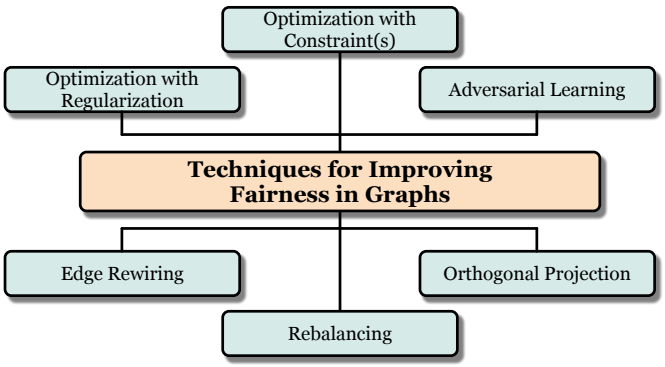


Fig. 3: A taxonomy of the commonly used techniques to improve fairness in graph mining.

works have been proposed to measure such biases from knowledge graph embeddings to fulfill social fairness. As an example, given the embeddings of head entities representing human and sensitive relations (e.g., gender and race), Fisher et al. [48] employed the prediction accuracy on tail entities (e.g., female/male when gender is the sensitive relation) as a bias metric. Here a low accuracy indicates that most sensitive information has been removed from the embeddings of these head entities. Meanwhile, social fairness in profession prediction can also be quantitatively measured through embedding perturbations [49]. Specifically, each human entity embedding is first perturbed to be more likely to have a certain value of the sensitive feature (e.g., gender, race, and occupation). Such perturbed embedding is then used to predict the profession corresponding to this human entity. The probability difference of having a certain profession between the perturbed and unperturbed embedding indicates the fairness level of the profession prediction. Here a larger difference implies a higher level of unfairness.

Path Diversity Fairness. Another fairness notion in knowledge graphs is *Path Diversity Fairness* [52], which is defined on *Meta-Paths*. Generally, a meta-path connects different object types (nodes) with relations (edges) on knowledge graphs [140]. Path diversity fairness requires that the distributions of meta-paths (typically defined by counting the number of paths for different meta-paths) should be similar across different sensitive subgroups [52]. Fu et al. [52] adopted Simpson's Index of Diversity (SID) [133] as the metric for unfairness.

Popularity Fairness. *Popularity Fairness* is also a critical fairness notion in knowledge graphs, and the popularity of each entity node is defined as its degree in the knowledge graph [8]. Generally, in knowledge graph completion tasks (i.e., predicting relations between entities or predicting a tail entity given head entity and a query relation), if the prediction accuracy is uniformly distributed w.r.t. entity node degrees, popularity fairness is then achieved.

4 Techniques for Improving Fairness

In this section, we introduce existing techniques for improving fairness in graph mining algorithms. Generally, these techniques can be divided into six categories, namely optimization with regularization, optimization with constraint(s), rebalancing, adversarial learning, edge rewiring, and orthogonal projection. We present the taxonomy of

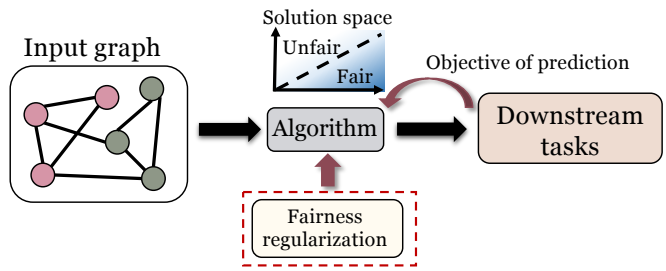


Fig. 4: The pipeline of optimization with regularization. The fairness regularization encourages the optimization result to stay in the area with a higher level of fairness (i.e., more blue) of the solution space.

techniques in Fig. 3. For the techniques in each category, we introduce how they promote application-agnostic and application-specific (if applicable) fairness. The surveyed literature is summarized in Table 2.

4.1 Optimization with Regularization

Optimization with regularization is a straightforward strategy to relieve unfairness in graph mining algorithms, and thus it is widely used in many existing works. The basic idea of regularization is to add an extra term to the objective of model utility to promote the fairness level of the algorithm output. Formally, the total objective $\mathscr{L}$ can be given as

$$\mathscr{L} = \mathscr{L}_{\text{utility}} + \lambda \mathscr{L}_{\text{fair}}, \tag{29}$$

where $\mathscr{L}_{\text{utility}}$ and $\mathscr{L}_{\text{fair}}$ are the objectives of utility and fairness, respectively; λ controls the effect of the fairness regularization. We present a basic pipeline in Fig. 4.

4.1.1 *Improving Group Fairness*

Algorithm Output-Based Regularization. Generally, the regularization enforcing statistical parity or equal opportunity is usually defined based on the algorithm output. For example, when the sensitive feature is binary, Zeng et al. [171] defined the regularization terms to enforce statistical parity and equal opportunity in node classification tasks. Specifically, the regularization for statistical parity is

$$\mathscr{L}_{sp} = \sum_{j=1}^{c} \left(\frac{\sum_{v_i \in \mathcal{V}_0} P(\hat{Y} = j \mid v_i)}{|\mathcal{V}_0|} - \frac{\sum_{v_i \in \mathcal{V}_1} P(\hat{Y} = j \mid v_i)}{|\mathcal{V}_1|} \right)^2, \tag{30}$$

and the regularization to enforce equal opportunity is

$$\mathscr{L}_{eo} = \sum_{j=1}^{c} \left(\frac{\sum_{v_i \in \mathcal{V}_0^j} P(\hat{Y} = j \mid v_i)}{|\mathcal{V}_0^j|} - \frac{\sum_{v_i \in \mathcal{V}_1^j} P(\hat{Y} = j \mid v_i)}{|\mathcal{V}_1^j|} \right)^2. \tag{31}$$

Here, $\mathcal{V}_0$ and $\mathcal{V}_1$ are the node sets for the two sensitive subgroups ($S = 0$ and $S = 1$), respectively; $\mathcal{V}_0^j$ and $\mathcal{V}_1^j$ are the sets of nodes that are predicted as class j in the two sensitive subgroups, respectively. Other recent works also follow a similar idea to design their regularizations for group fairness [50], [110], [156], [167].

TABLE 2: Surveyed publications on graph mining algorithms categorized by the fairness notions and techniques.

Fairness Notions	Techniques	Publications	Downstream Tasks
Application-Agnostic			
Group	Optimization with Regularization	[71]	Recommendation
	Optimization with Regularization	[67]	Node classification
	Optimization with Regularization	[173]	Node classification
	Optimization with Regularization	[3]	Node classification
	Optimization with Regularization	[24]	Link prediction
	Optimization with Regularization	[167]	Recommendation
	Optimization with Regularization	[88]	Node classification
	Optimization with Regularization	[44]	Node classification
	Optimization with Regularization	[110]	Node classification
	Optimization with Regularization	[50]	Node classification
	Optimization with Regularization	[175]	Recommendation
	Optimization with Regularization	[52]	Recommendation
	Optimization with Regularization	[156]	Link prediction
	Optimization with Regularization	[83]	Recommendation
	Optimization with Regularization	[87]	Link prediction
	Optimization with Regularization	[21]	Recommendation
	Optimization with Constraint(s)	[80]	Graph clustering
	Optimization with Constraint(s)	[171]	Node classification
	Optimization with Constraint(s)	[45]	Influence maximization
	Optimization with Constraint(s)	[123]	Influence maximization
	Optimization with Constraint(s)	[124]	Influence maximization
	Optimization with Constraint(s)	[5]	Influence maximization
	Rebalancing	[128]	Link prediction
	Rebalancing	[25]	Link prediction
	Rebalancing	[171]	Node classification
	Rebalancing	[46]	Recommendation
	Rebalancing	[122]	Recommendation
	Rebalancing	[149]	Node ranking
	Rebalancing	[77]	Influence maximization, link prediction, and node classification
	Rebalancing	[146]	Influence maximization
	Rebalancing	[138]	Influence maximization
	Rebalancing	[148]	Influence maximization
	Rebalancing	[82]	Node classification
	Rebalancing	[37]	Link prediction
	Adversarial	[18]	Recommendation
	Adversarial	[38]	Node classification
	Adversarial	[78]	Influence Maximization
	Adversarial	[164]	Recommendation
	Adversarial	[159]	Recommendation
	Adversarial	[78]	Influence maximization
	Edge rewiring	[91]	Node classification
	Edge rewiring	[82]	Node classification
	Edge rewiring	[65]	Topology debiasing
	Edge rewiring	[40]	Node classification
	Edge rewiring	[82]	Node classification
	Edge rewiring	[136]	Node classification
	Orthogonal projection	[114]	Node classification and recommendation
	Orthogonal projection	[113]	Node classification and recommendation
	Orthogonal projection	[171]	Node classification
Individual	Optimization with Regularization	[39]	Node classification and link prediction
	Optimization with Regularization	[44]	Node classification
	Optimization with Regularization	[88]	Node classification
	Optimization with Regularization	[73]	Node ranking, node classification and graph clustering
	Optimization with Constraint(s)	[59]	Graph clustering
	Edge rewiring	[87]	Link prediction
Degree-Related	Rebalancing	[145]	Node classification
	Rebalancing	[75]	Node classification
	Rebalancing	[47]	Influence maximization
Application-Specific			
Popularity	Optimization with Regularization	[72]	Recommendation
	Optimization with Regularization	[157]	Recommendation
	Optimization with Regularization	[1]	Recommendation
	Optimization with Regularization	[33]	Recommendation
	Optimization with Regularization	[174]	Recommendation
	Optimization with Regularization	[54]	Recommendation
	Adversarial	[84]	Recommendation
	Edge rewiring + Adversarial	[101]	Link prediction
Provider	Optimization with Regularization	[99]	Recommendation
	Rebalancing	[16]	Recommendation
	Rebalancing	[117]	Recommendation
User	Optimization with Regularization	[97]	Recommendation
	Optimization with Regularization	[93]	Recommendation
	Rebalancing	[76]	Recommendation
	Rebalancing	[105]	Recommendation
Marketing	Optimization with Regularization	[154]	Recommendation
Social	Optimization with Regularization	[49]	Knowledge graph embedding learning
	Adversarial	[48]	Triple prediction
	Adversarial	[8]	Link prediction

Network Topology-Based Regularization. Regularization based on the network topology is proved to be effective in promoting group fairness. For instance, feature propagation is a common operation to model the dependency between neighboring nodes in graph mining [67], [131]. However, if the graph topology is biased, the propagated features also tend to be biased [40]. To tackle such problem, Jiang et al. [67] achieved a less biased feature propagation through a fairness-aware regularization. Specifically, given two sensitive subgroups, the regularization is formally given as

$$\mathscr{L}_{\text{fair}} = \|\mathbf{\Delta}_s \text{softmax}(\hat{\mathbf{X}})\|_1. \tag{32}$$

Here softmax$(\cdot)$ is the softmax function; $\hat{\mathbf{X}} \in \mathbb{R}^{n\times d}$ is the node feature matrix after propagation; $\mathbf{\Delta}_s \in \{1, -1\}^{1\times n}$ is the element-wise indicator for the sensitive subgroup membership, which is formulated as

$$\mathbf{\Delta}_s = \frac{\mathbb{1}_{=1}(\mathbf{s})}{\|\mathbb{1}_{=1}(\mathbf{s})\|_1} - \frac{\mathbb{1}_{=0}(\mathbf{s})}{\|\mathbb{1}_{=0}(\mathbf{s})\|_1}, \tag{33}$$

where $\boldsymbol{s} \in \{0, 1\}^n$ is the sensitive feature vector for all nodes; $\mathbb{1}_{=0}(\cdot)$ and $\mathbb{1}_{=1}(\cdot)$ is the element-wise indicator function for 0 and 1 entries, respectively. As such, $\mathbf{\Delta}_s$ is a vector where entries corresponding to members in the two sensitive subgroups are 1 and -1 normalized by their subgroup size, respectively. This regularization encourages the average feature values after propagation to be similar between the two subgroups at each dimension. Additionally, regularization based on the network topology is also widely employed in link prediction. As an example, Buyl et al. [24] proposed a fairness regularization term for the link prediction task. They first defined a set of probabilistic graph models that are fair w.r.t. demographic parity and equal opportunity. Then they measured the KL-divergence between the predicted edge distribution (by the link prediction model) and the edge distribution determined by its closest graph model. Such KL-divergence is employed as the regularization term, and minimizing it encourages the predicted edge distribution to be close to a pre-defined fair edge distribution.

Node Embedding-Based Regularization. Regularization based on node embeddings is another common approach to improve group fairness. For example, Lahoti et al. [88] utilized the total Euclidean distance of all embedding pairs spanning across different sensitive subgroups as a regularization term, which encourages the node embeddings in different sensitive subgroups to be similar. Apart from that, in binary sensitive feature scenarios, the distribution distance of node embeddings between the two sensitive subgroups is also an effective regularization term that helps to fulfill group fairness [44], [110].

4.1.2 Improving Individual Fairness

Algorithm Output-Based Regularization. Regularization can also be adopted to improve individual fairness for graph mining algorithms. A basic desideratum here is to employ regularization terms to reduce the output difference between nodes that are similar, which is consistent with the intuition "to treat similar individuals similarly". For example, Kang et al. [73] leveraged an oracle similarity matrix $\mathbf{S}$ to indicate the similarity between individuals, and the Laplacian matrix $\mathbf{L_S}$ for $\mathbf{S}$ can then be derived. The total variation of the output matrix $\hat{\mathbf{Y}}$ w.r.t. $\mathbf{S}$ is used as the regularization term, which can be formulated as $\text{Tr}(\hat{\mathbf{Y}}^\top \mathbf{L_S} \hat{\mathbf{Y}})$. With such a regularization, the algorithm yields similar outputs for similar nodes, which aligns with the definition of individual fairness in Section 3.2.1. Lahoti et al. [88] followed a similar idea. In their work, the similarity between nodes is derived from both node features and human knowledge. As another example, Dong et al. [39] formulated a regularization term based on node rankings to promote individual fairness in GNNs. Specifically, for each individual, a ranking list is first derived based on the input similarity scores between itself and other individuals. Similarly, another ranking list can also be derived based on the GNN output similarity between this individual and others. Such a regularization encourages the two ranking lists for each individual to be as similar as possible, which promotes the node ranking-based fairness.

Node Embedding-Based Regularization. Promoting the level of group fairness based on node embedding distributions also helps to impose individual fairness [43], [44]. Fan et al. [44] empirically proved that the Wasserstein distance between node embedding distributions across different sensitive subgroups can be utilized as an effective regularization to improve fairness at both group and individual level in node classification tasks.

4.1.3 Improving Counterfactual Fairness

The intuition of counterfactual fairness on graphs is that the prediction of each individual (node) should be the same on the factual data and counterfactuals (in counterfactuals, the values of nodes' sensitive feature have been changed). Based on such intuition, many studies have been devoted in recent years. Among them, Agarwal et al. proposed NIFTY [3], which generates the graph counterfactuals by flipping the sensitive feature values for all nodes while keeping everything else unchanged. A regularization in the loss function is then introduced to encourage the node embeddings learned from the factual graph and its counterfactual to be the same. At a high level, the regularization for counterfactual fairness can be expressed as $E[D(\mathbf{z}_i, \mathbf{z}'_i)]$, where $\mathbf{z}_i$ and $\mathbf{z}'_i$ are embeddings of node v_i learned based on the factual graph and its counterfactual, respectively; $D(\cdot, \cdot)$ is a distance metric. A following work [104] proposed a similar regularization term. The main difference is that more causal relations are considered to generate graph counterfactuals.

4.1.4 Improving Fairness in Recommender Systems

In recommender systems, regularization is a commonly used technique to fulfill popularity fairness. As an example, in online shopping platforms, the number of feedback actions an item receives generally represents how popular this item is. Based on such intuition, Zhu et al. [174] proposed to formulate a regularization as

$$\mathscr{L}_{fair} = \text{Corr}_{\text{P}}(\hat{\mathbf{r}}_+, \mathbf{p}_+). \tag{34}$$

Here $\hat{\mathbf{r}}_+$ denotes the vector of predicted relevance scores for positive user-item pairs; $\mathbf{p}_+$ represents the vector of the feedback number received by the corresponding items in user-item pairs; $\text{Corr}_{\text{P}}(\cdot, \cdot)$ is the Pearson correlation function. By regularizing the correlation between $\hat{\mathbf{r}}_+$ and $\mathbf{p}_+$,

the effect that popular items tend to receive higher relevance scores can be relieved. Various other works also employed regularization terms based on the popularity of items to achieve popularity fairness [1], [33], [54], [72], [157].

Regularization is also a widely used technique to promote user fairness in recommender systems. For example, Lin et al. [97] pointed out that all users should receive recommendations with equal quality. Correspondingly, the difference of recommendation quality (measured by the relevance scores of recommended items to users) between every two individuals is summed up and formulated as the regularization term. Besides, Li et al. [93] defined a regularization term by considering the recommendation quality discrepancy between active users and inactive ones.

Furthermore, regularization can also be adopted to promote other types of application-specific fairness. For example, to achieve provider fairness, the provider diversity of recommended items is often employed as a regularization [16], [99]. To fulfill marketing fairness, the variances of errors between recommendations to identity-consistent and identity-inconsistent users (see definitions in Section 3.5.1) are first computed, and their discrepancy is then adopted as a regularization [154].

4.1.5 Improving Fairness in Knowledge Graphs

Regularization is usually formulated in knowledge graph embedding learning to fulfill fairness for human entities. For example, Fisher [49] proposed to relieve social bias through regularization. Specifically, a prediction on the sensitive feature value is made for each human entity based on the learned embeddings. The KL-divergence between the predicted value distribution and uniform distribution over all possible sensitive feature values is then defined as the regularization term. Generally, a smaller KL-divergence indicates that the embeddings provide less information about the sensitive feature, which implies a higher level of fairness.

4.2 Optimization with Constraint(s)

Optimization with constraint(s) serves as a critical technique to fulfill fairness in graph mining. Generally, such fairness constraint(s) reduce the feasible set size of the corresponding optimization problem to exclude unfair solutions. We present a conceptual formulation of optimization problems with fairness constraint(s) as:

$$\begin{aligned} \min \quad & \mathscr{L}_{\text{utility}}, \\ \text{subject to} \quad & \text{certain fairness constraint(s)} \end{aligned} \tag{35}$$

We present a pipeline of optimization with constraint(s) in Fig. 5. Different from regularization, optimization with constraint(s) requires that the solution should only be in the fair area (i.e., the blue area) of the solution space.

4.2.1 Improving Group Fairness

Optimization with constraint(s) is widely leveraged in influence maximization to ensure the influence propagates to different sensitive subgroups in a fair manner [5], [45], [124].

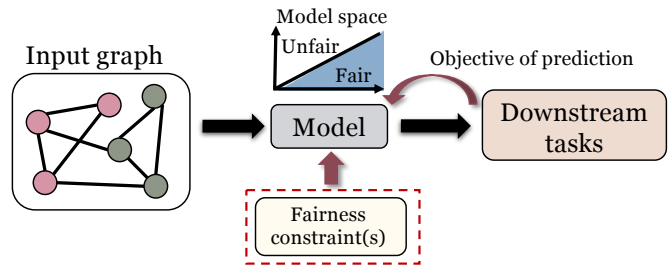


Fig. 5: The pipeline of optimization with constraint(s). The formulated hard constraint for fairness restricts the solution to be in the fair area (i.e., blue area) of the solution space.

Generally, the fairness-constrained influence maximization problem is formulated as

$$\begin{aligned} \max \quad & \underbrace{\sum_{i=1}^{H} u_i(\mathcal{A})}_{\text{Expected number of influenced nodes}} \\ \text{subject to} \quad & \underbrace{|\mathcal{A}| \leq K}_{\text{Bound of seed set size}} \\ \text{and} \quad & \underbrace{M(\mathcal{A}, u_1, ..., u_H, \mathcal{V}_1, ..., \mathcal{V}_H) \leq \beta_{\text{unfair}}}_{\text{Fairness constraint}} \end{aligned} \tag{36}$$

Here $u_i(\mathcal{A})$ is the utility (i.e., the percentage of influenced nodes) of the i-th sensitive subgroup based on the seed node set $\mathcal{A}$; K represents the budget for seed set size; function M outputs the unfairness level of the influence maximization algorithm according to certain unfairness metric; β_{unfair} denotes the maximum acceptable threshold for the influence maximization unfairness. It is worth noting that the constrained optimization problem given by Eq. (36) has been proved to be an NP-hard problem [5]. Hence the optimization problem introduced above is usually transformed into its surrogate problem and solved in a heuristic manner.

Aside from influence maximization, optimization with constraint(s) is also used to fulfill fairness in other graph mining scenarios such as graph clustering. As an example, Matthäus et al. [80] defined a fairness-aware constraint for spectral clustering to ensure that each sensitive subgroup is proportionally represented by each cluster. Formally, assume nodes in the input graph are divided into H sensitive subgroups, and $\mathcal{V}_i$ is the node set of the i-th subgroup. K and $\mathcal{C}_k$ are the cluster number and the node set of the k-th cluster, respectively. The constraint is given as

$$\forall i \in \{1, ..., H\} \text{ and } \forall k \in \{1, ..., K\}, \frac{|\mathcal{V}_i \cap \mathcal{C}_k|}{|\mathcal{C}_k|} = \frac{|\mathcal{V}_i|}{|\mathcal{V}|}. \tag{37}$$

4.2.2 Improving Individual Fairness

Optimization with constraint(s) is also a popular technique to fulfill individual fairness. For example, Gupta et al. [59] proposed to achieve individual fairness in graph clustering via optimization constraints. As introduced in Section 3.2.3, individual fairness in graph clustering requires that for each node in the graph, its neighbors should be proportionally assigned to different clusters. To this end, the corresponding constraint is formulated as $\forall k \in \{1, ..., K\}, \frac{1}{|\mathcal{C}_k|} |\{v_j : \mathbf{A}_{ij} = 1 \wedge v_j \in \mathcal{C}_k\}| =$

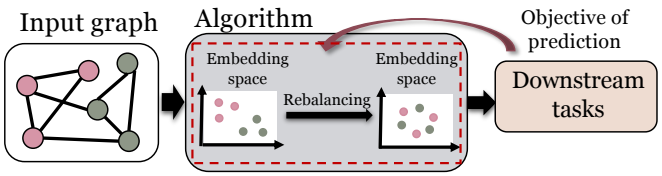


Fig. 6: The pipeline of rebalancing-based approaches. Generally, rebalancing aims to make certain characteristics to be as balanced as possible between the advantaged and disadvantaged individuals.

$\frac{1}{|\mathcal{V}|}|\{v_j : \mathbf{A}_{ij} = 1\}|$ for node v_i, where $1 \leq i \leq n$. To be more concise, such constraint can be further formulated as

$$\mathbf{A}(\mathbf{I} - \mathbf{1}\mathbf{1}^\top/n)\mathbf{H} = \mathbf{0}, \tag{38}$$

where $\mathbf{1} \in \{1\}^{n\times 1}$ and $\mathbf{0} \in \{0\}^{n\times K}$. For $k \in \{1, ..., K\}$, matrix $\mathbf{H}$ is defined as

$$\mathbf{H}_{i,k} = \begin{cases} 1/\sqrt{|\mathcal{C}_k|}, & \text{if } v_i \in \mathcal{C}_k, \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

4.3 Rebalancing

Rebalancing aims to reduce the distribution difference of certain properties (e.g., the appearance rate of a node in random walks and frequency of recommendations of an item) between advantaged and disadvantaged nodes. We present a basic pipeline in Fig. 6.

4.3.1 Improving Group Fairness

Network Topology-Based Rebalancing. A number of works adopt the rebalancing strategy to promote group fairness based on the network topology. As an example, Rahman et al. [122] proposed Fairwalk to achieve fair node embedding learning, where the node appearance frequency in random walks (generated by *node2vec* [56]) is balanced between minority and majority groups. However, when a walk is choosing its next step in FairWalk, only one-hop neighbors of the node at the current step are considered. Such a strategy lacks long-term rebalancing and thus could fail when all nodes up to several hops away from the current step have the same membership (i.e., the current step has a homogeneous surrounding). To consider long-term rebalancing, Khajehnejad et al. [77] proposed CrossWalk, which extends the rebalancing range to the whole walk by assigning larger transition probabilities to nodes that are closer to the sensitive groups' topological peripheries. Compared with FairWalk, CrossWalk further promotes the diversity of node membership by encouraging walks to avoid being stuck in homogeneous surroundings. On heterogeneous graphs, topology-based rebalancing is also adopted for fair embedding learning. For example, during meta-path generation, the probabilities of selecting nodes from different sensitive subgroups are rebalanced in [171]. Specifically, rebalancing is achieved via selecting nodes in disadvantaged subgroups with a higher probability. Such a strategy enforces a fair appearance rate for nodes from different subgroups in the generated meta-paths. Another example is the fairness-aware PageRank [149]. Generally, the node importance vector given by PageRank [19] is derived based on the transition matrix and the jump vector. However, they could be biased due to the imbalanced size of different sensitive subgroups. By rebalancing the transition (jumping) probabilities across different sensitive subgroups, group fairness can be achieved in the fairness-aware PageRank algorithm. Furthermore, in link prediction tasks, Saxena et al. [128] rebalanced the number of intra-group and inter-group links, which also effectively enforces group fairness.

Node Sampling/Generation-Based Rebalancing. Rebalancing can also be achieved via node sampling or node generation. Generally, both approaches can rebalance the node number between different sensitive subgroups. For node sampling, Kose et al. [82] pointed out that if a GNN model is trained on a sampled subgraph with balanced populations from different sensitive subgroups, its predictions tend to be with a higher level of group fairness. For node generation, Current et al. [37] proposed to generate pseudo nodes and reweight edges in the input graph of GNNs to encourage balanced information propagation in different sensitive subgroups. Such modifications on the input graph are jointly optimized with the GNN model parameters.

Information Flow-Based Rebalancing. Information flow-based rebalancing techniques are commonly adopted to achieve fair influence maximization. For example, Stoica et al. [138] proposed *Parity Seeding*, which is achieved by setting different seed number budgets for different sensitive subgroups. In this way, the flow of influence originating from seed nodes is rebalanced across different subgroups. Besides, Tsang et al. [148] rebalanced the selection of seed nodes to improve the lowest ratio of influenced nodes among all sensitive subgroups.

4.3.2 Improving Degree-Related Fairness

Degree-related fairness can also be achieved via rebalancing techniques. For example, in the message-passing process of traditional GNNs, nodes with low degrees usually benefit less (compared with nodes with high degrees) from the information propagation due to their sparse connections [75]. Tang et al. [145] proposed to rebalance the labeled nodes across the graph. Specifically, pseudo labels are generated to improve the probability of labeled nodes appearing in the neighborhood of low-degree nodes. In this way, more supervision information can be accessed by those low-degree nodes through the given network topology. Such a rebalancing strategy has been proved to be effective in improving the node classification accuracy for low-degree nodes. Besides, Kang et al. [75] pointed out that a critical source of degree-related unfairness in GNNs is the gradient of learnable weight parameters w.r.t. the objective function. In particular, it has been proved that high-degree nodes tend to exert a more significant influence on the gradient of the learnable weight matrix, which is the reason why GNNs favor high-degree nodes. To handle this problem, a doubly stochastic adjacency matrix (the rows and columns sum up to 1) of the GNN input network is defined and employed as GNN input. Such a strategy rebalances the influence of each node to the learnable weight matrix during optimization, which helps to enforce degree-related fairness.

Additionally, in information maximization, Fish et al. [47] designed a *Social Welfare Function* to measure the difficulty for the low-degree nodes to get access to the information originating from the seed nodes. Based on the welfare function, the probability that the seed nodes reach

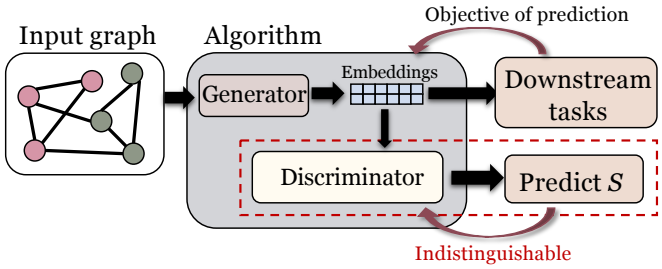


Fig. 7: The pipeline of adversarial learning-based approaches. The information of sensitive feature(s) is removed from the embeddings if the discriminator cannot accurately predict sensitive feature values.

out and influence those low-degree nodes is promoted. Such a strategy helps to enable nodes with high and low degrees to receive a more balanced amount of influence.

4.3.3 Improving Fairness in Recommender Systems

Upsampling is a common rebalancing approach in recommender systems. For example, in terms of provider fairness, Boratto et al. [16] proposed to upsample interactions between users and items from minority providers. A similar rebalancing idea is also adopted by Gourab et al. [117], where copies of products are made to improve the exposure of items from minority providers.

Rebalancing can also be leveraged to improve user fairness. For example, in group recommendation, items are recommended to a group of users. To ensure that the preference of each user is proportionally represented by the recommended items, Malecek et al. [105] proposed to enforce a cap on the recommendation relevance score summation for each user to rebalance the recommendation quality. In this way, the phenomenon that some users in a group may be under-represented and receive unsatisfying recommendation results can be eliminated. Similar rebalancing approaches have also been applied to rebalance item ratings given by users from different sensitive subgroups to achieve a higher level of user fairness [46].

4.4 Adversarial Learning

The adversarial learning-based framework includes a generator and a discriminator. The generator outputs node embeddings or probabilistic predictions, while the discriminator aims to predict the sensitive feature values based on the generator output. The basic rationale here is to play a min-max game between the generator and discriminator. When the discriminator fails to predict sensitive feature values, the generator output is regarded as decoupled from the sensitive feature(s) [18]. We present a basic pipeline of adversarial learning-based approaches in Fig. 7.

4.4.1 Improving Group Fairness

Adversarial learning is a popular strategy for learning node embeddings that are fair in terms of group fairness. For example, Bose et al. [18] leveraged a discriminator to predict the value of sensitive feature(s) based on learned node embeddings, while the generator aims to generate embeddings that are indistinguishable w.r.t. sensitive feature(s). This idea is also followed by many other works [38], [98], [159], [164] to filter out the information of sensitive features from the learned node embeddings. In [78], Khajehnejad et al. proposed to learn node embeddings based on adversarial learning to promote group fairness for influence maximization. With the learned embeddings, seed nodes are selected based on embedding clustering: the nodes nearest to the centroid of each cluster are selected. Considering that the information of sensitive features has been removed from the learned embeddings, the seed selection is regarded as fair.

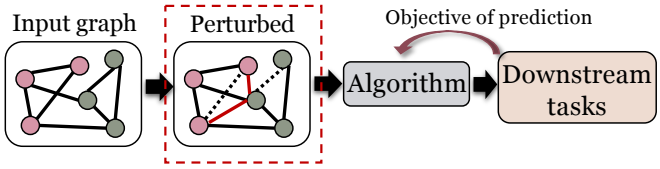


Fig. 8: The pipeline of edge rewiring-based approaches. Edges in the input graph data is rewired (i.e., removed from linked node pairs and added to non-linked node pairs) to obtain fair algorithm output in downstream tasks.

4.4.2 Improving Fairness in Knowledge Graphs

Adversarial learning can also be adopted to improve social fairness in knowledge graph embedding learning. Arduini et al. [8] proposed to leverage a sensitive information filter to remove social bias from the embeddings of human entities. In their paper, the sensitive information filter also plays a min-max game with a discriminator, such that the entity embeddings are decoupled from the sensitive feature(s).

4.5 Edge Rewiring

Biases exhibited in the node embeddings and algorithm predictions could also be attributed to the biased network topology. In this regard, modifying the graph topology through edge rewiring is a common debiasing strategy. We present a basic pipeline in Fig. 8.

4.5.1 Improving Group Fairness

Information Flow-Based Rewiring. In most graph mining algorithms, there are information flows from nodes to nodes [32], [79], [152], [165]. An intuitive idea to mitigate group unfairness is to modify the graph topology to make such information flows as fair as possible. For instance, Jalali et al. [65] proposed *Information Unfairness Score* based on the information flows. Specifically, given several groups of nodes, the information unfairness score depicts the largest distribution difference of the probabilistic accessibility between two node groups. To obtain a fair graph topology, edges are rewired in a greedy manner to maximally reduce the information unfairness score. Additionally, in GNNs, the information aggregation operation is found to introduce bias from the biased network topology to the learned node embeddings [40], [67], [81], [82], [91]. Dong et al. [40] proposed to perform edge rewiring for fair node embedding learning. Specifically, the Wasserstein distance between the node embedding distributions from two sensitive subgroups is minimized by learning a less biased (weighted) graph adjacency matrix. The learned weights in the adjacency matrix are converted into binary values according to a pre-assigned threshold for edge rewiring. Similar edge rewiring ideas are also adopted by other works. For

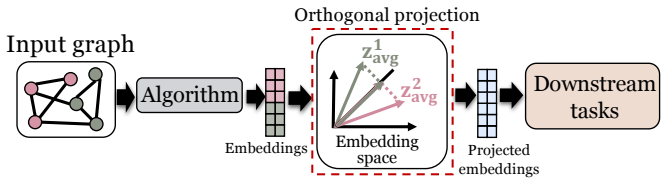


Fig. 9: The pipeline of orthogonal projection-based approaches. The embeddings of individuals who are discriminated against and those who are not are projected onto the same hyperplane in the embedding space.

example, Li et al. [91] proposed to optimize the adjacency matrix to minimize the expected probability difference of being connected between inter- and intra-group node pairs in link prediction tasks.

Edge Sampling-Based Rewiring. Edges can also be sampled in a probabilistic way to improve group fairness. For example, Spinelli et al. [136] pointed out that nodes within the same sensitive subgroup tend to be linked together on homogeneous graphs. The dominance of these intra-group edges could lead to bias in embedding learning. To tackle this issue, Spinelli et al. proposed a debiasing approach named FairDrop, where more intra-group edges than inter-group edges are removed according to a probabilistic edge-masking matrix. Similar probabilistic edge removing approaches are also adopted by other works such as [82].

4.5.2 Improving Individual Fairness

In terms of individual fairness, the edge rewiring strategy encourages similar individuals to share similar topological characteristics. For example, algorithms based on a biased network topology tend to yield biased results in downstream tasks [40], [87]. To tackle this issue, an edge rewiring strategy is introduced by Laclau et al. [87] to achieve a fair topology for downstream tasks. Specifically, a matrix $\mathbf{S}$ is first given to indicate the pair-wise node similarity. To optimize the network topology, an optimization problem is then formulated to encourage similar nodes to have highly overlapped neighboring node sets after edge rewiring. Downstream tasks are proved to benefit from the rewired network topology in terms of individual fairness.

4.5.3 Improving Fairness in Recommender Systems

In recommender systems, edge rewiring can be leveraged to tackle the well-known filter bubble problem. For example, Masrour et al. [101] proposed an extended modularity score (as presented in Section 3.5.1) of the graph as a popularity fairness indicator. Based on the obtained link prediction results, a proportion of links are rewired in a greedy manner to promote the modularity score, which helps to achieve popularity fairness.

4.6 Orthogonal Projection

To decorrelate the learned embeddings from the sensitive feature(s), making the learned node embeddings orthogonal to the corresponding sensitive feature(s) is an effective approach. This is usually achieved through projecting the node embeddings onto a hyperplane orthogonal to the direction of the sensitive features. We present a basic pipeline in Fig. 9. Compared with other approaches, it provides a theoretical guarantee that the node embeddings are uncorrelated with the sensitive feature(s) [114].

4.6.1 Improving Group Fairness

Orthogonal projection is an effective approach to improve group fairness for graph embeddings. In [171], Zeng et al. defined *Bias Direction*. Based on such, the node embeddings are projected onto a hyperplane orthogonal to the bias direction. Specifically, for the i-th sensitive subgroup, we obtain an averaged unit node embedding as

$$\mathbf{z}^i_{\text{avg}} = \frac{\mathbf{z}_1 + \mathbf{z}_2 + ... + \mathbf{z}_{|\mathcal{V}_i|}}{\|\mathbf{z}_1 + \mathbf{z}_2 + ... + \mathbf{z}_{|\mathcal{V}_i|}\|_2}, \tag{40}$$

where for a binary sensitive feature, $i \in \{1, 2\}$; $\mathbf{z}_j$ ($j \in \{1, 2, ..., |\mathcal{V}_i|\}$) denotes the learned embedding of node v_j. The unit vector in the bias direction is defined as

$$\mathbf{z}_{\text{bias}} = \frac{\mathbf{z}^1_{\text{avg}} - \mathbf{z}^2_{\text{avg}}}{\|\mathbf{z}^1_{\text{avg}} - \mathbf{z}^2_{\text{avg}}\|_2}. \tag{41}$$

Generally, if all node embeddings are projected onto a hyperplane that is orthogonal to $\mathbf{z}_{\text{bias}}$, then the component of the projected node embeddings in the direction of $\mathbf{z}_{\text{bias}}$ is zero, i.e., the sensitive information is decorrelated from the learned node embeddings. Correspondingly, for node v_j, the projected embedding is formulated as

$$\mathbf{z}'_j = \mathbf{z}_j - < \mathbf{z}_j, \mathbf{z}_{\text{bias}} > \mathbf{z}_{\text{bias}}, \tag{42}$$

where $\mathbf{z}'_j$ is the projected embedding for node v_j, and $< \cdot, \cdot >$ is the inner product operator. Similarly, Palowitch et al. [114] proposed to learn topological embeddings by projecting the embedding onto a hyperplane orthogonal to the hyperplane of node features. This offers the theoretical guarantee that there will be no correlation between the potentially biased node features and topological embeddings. However, it is worth noting that orthogonal projection only guarantees that the embeddings and sensitive feature(s) are uncorrelated, while how to exclude non-linear dependency between them remains under-explored.

5 Benchmark Datasets

We summarize the datasets used by the surveyed literature in Table 3. In general, we group these datasets into four categories w.r.t. the type of networks, including social networks (C1), recommendation-based networks (C2), academic networks (C3), and other types of networks (C4). For each type of network, we present the fairness notions that can be studied on these datasets and the basic dataset statistics, including the number of nodes, the number of edges, the number of node features, the semantic meaning of sensitive feature(s), and the number of sensitive subgroups. Furthermore, we also list the papers that used these datasets for fairness studies in graph mining. We collect these open-source datasets in https://github.com/yushundong/Graph-Mining-Fairness-Data.

6 Research Challenges

Here we introduce the limitations of current research, pressing challenges, and open questions for future advances.

Formulating Fairness Notions. Discrimination could exist in diverse forms in graph mining. Correspondingly, different types of fairness notions should be formulated towards a comprehensive understanding of bias and discrimination

TABLE 3: Collection of benchmark datasets. "CI, C2, C3, C4" refer to social networks, recommendation-based networks, academic networks, and other types of networks, respectively. Different versions exist for datasets marked with "*", and only the statistics corresponding to the most representative version are presented.

	Dataset	Fairness Goal(s)	# nodes	# edges	# features	S ($\|S\|$)	Works Citation
C1	Facebook* [63]	group, individual	1,034	26,749	224	gender (2)	[5], [24], [37], [39], [44], [73], [78], [80] [83], [87], [101], [128], [146], [171]
	Pokec* [142]	group	1,632,803	30,622,564	59	region (2), gender (2)	[38], [40], [50], [67], [82], [110]
	Twitter* [107]	group	81,306	1,768,149	1,364	political opinion (2)	[77], [83], [149]
	Lastfm [27]	group, provider	49,900	518,647	-	gender (2), age (3)	[117], [159], [164]
	Oklahoma97 [126]	group	3,111	73,230	8	gender (2)	[91]
	UNC28 [126]	group	4,018	65,287	8	gender (2)	[91]
	Google+ [107]	popularity	4,938	547,923	5	-	[101]
	Epinion [144]	popularity	8,806	157,887	-	-	[1], [174]
	Filmtrust [58]	provider	3,579	35,494	-	-	[99]
	Ciao [144]	popularity	7,317	85,205	-	-	[174]
C2	Amazon* [90]	group, marketing, social	334,863	925,872	-	product category (4)	[52], [83], [93], [154], [175]
	Yelp* [175]	group, social	12,683	211,721	14	food genre (4)	[175]
	MovieLens-100K [62]	group, popularity	2,625	100,000	12	gender (2), age (7), occupation (21)	[24], [25], [54], [114]
	MovieLens-1M [62]	group, individual, popularity, provider, social, user	10,000	1,000,000	11	gender (2), age (7), occupation (21)	[1], [18], [21], [33], [46], [54] [71], [76], [97], [99], [105] [159], [164], [171], [174], [175]
	MovieLens-20M [62]	popularity	165,000	20,000,000	6	-	[84], [157]
C3	Citeseer [130]	group, degree-related	3,327	4,732	3,703	topic (6)	[37], [82], [91], [136], [145]
	Cora [130]	group, degree-related	2,708	5,429	1,433	topic (7)	[37], [82], [91], [136], [145]
	Pubmed [130]	group, degree-related	19,717	44,338	500	topic (3)	[37], [82], [91], [136], [145]
	DBLP* [143]	group, individual	3,980	6,965	-	continent (5), gender (2)	[25], [65], [87], [136], [138], [149]
C4	German [9]	group, couterfactual	1,000	21,742	27	gender (2)	[3], [40], [173]
	NBA [38]	group	403	10,621	96	country (2)	[38], [67]
	Recividism [69]	group, individual couterfactual	18,876	311,870	18	race (2)	[3], [40], [44], [104], [173]
	Credit [168]	group, couterfactual	30,000	1,421,858	13	age (2)	[3], [40], [104], [173]

in different real-world applications [23], [28], [134], [158], [169]. Although we have surveyed many fairness notions for graph mining, we need to admit that by no means are they complete, as other types of biases could also exist, depending on the needs of different real-world scenarios [160]. On the other hand, the definitions of different fairness notions on graphs could even be in conflict with each other [13], [160]. Therefore, designing a new fairness notion or choosing a set of existing non-conflicting fairness notions for particular graph mining algorithms and downstream applications remains an open question.

Fulfilling Multiple Types of Fairness. It should be noted that any type of bias is undesired in real-world applications. In this regard, there is an urgent need to promote multiple types of fairness at the same time. For example, group fairness and individual fairness can be promoted at the same time under certain scenarios [44], [88]. However, promoting multiple types of fairness at the same time is a non-trivial problem, as promoting one type of fairness may degrade several other types of fairness [13], [22]. Such a phenomenon can be more pronounced on graphs, which is resulted from the dependency between neighboring nodes. For example, in a social network, individuals with the same gender are more densely connected. In this case, individual fairness enforces the nodes in the same gender subgroup to be similar (e.g., similar embeddings). However, such a goal may lead to a larger discrepancy between gender subgroups, which adversely affects the level of group fairness. Therefore, properly addressing multiple unfairness issues in graph mining simultaneously is a pressing problem.

Balancing Model Utility and Algorithmic Fairness. For algorithms with fairness considerations, the utility such as prediction accuracy is usually sacrificed [30], [115], [161]. Such a trade-off between utility and fairness has been studied on i.i.d. data in recent years. To achieve a satisfying trade-off, a common strategy is to ensure the algorithm bearing Pareto optimality [106], [132], i.e., a state where either utility or fairness cannot be promoted without harming the other one. Graph mining algorithms also have the issue of utility-fairness trade-off [38], [39], [55]. For example, when the fairness-related regularization is added to the objective function of a specific graph analytical task, the solution of the regularized optimization problem often deviates from the solution that brings the best utility in the unregularized optimization problem. Additionally, in an adversarial learning-based framework, when the generator successfully fools the discriminator, some useful information may also be wiped out from the embeddings or predictions given by the generator. This could also degrade the model utility performance in downstream tasks. Hence it is critical to study how to achieve a trade-off between utility and fairness.

Explaining How Unfairness Arises. Although various debiasing strategies have been proposed to debias graph mining algorithms, systematically understanding how such unfairness arises in the underlying algorithm is also crucial. However, this problem can be challenging. A reason is that the exhibited unfairness is usually coupled with both the input graph and specific mechanisms in graph mining algorithms. For example, due to the message-passing mechanism in GNNs, the unfairness exhibited in the learned node embeddings can be attributed to the biased input graph topology [40]. Systematically explaining how unfairness

arises in various graph mining algorithms remains a critical issue to be addressed.

Enhancing Robustness of Algorithms on Fairness. In graph mining, enhancing the robustness of graph mining algorithms w.r.t. fairness is another urgent need. For instance, in learning-based algorithms, human annotators could provide biased supervision information for model training [20]. Besides, the algorithms may also be manipulated by malicious attackers to exhibit discrimination against a certain group of people [7], [135]. In both cases, the fairness level of the algorithm predictions can be dramatically lowered. Despite the significance of enhancing the robustness of algorithmic fairness, most existing studies are overwhelmingly devoted to i.i.d. data [108], [135], and cannot be directly grafted to the graph-structured data. In this regard, how to promote the robustness of the fairness aspect of graph mining algorithms deserves further investigation.

7 Conclusion

Graph mining has achieved remarkable success in a myriad of high-impact real-world applications. Nevertheless, due to the lack of fairness considerations, there has been an increasing societal concern that these algorithms may exhibit discrimination when they are exploited to make predictions and decisions. Over the years, many efforts have been made to define, measure, and promote fairness in graph mining. In this survey, we propose a novel taxonomy of fairness notions in graph mining research and systematically review existing fairness notions from different perspectives. Besides, we categorize and introduce existing techniques that promote fairness in graph mining. Furthermore, rich benchmark graph datasets are collected to facilitate future research advances in this area. Finally, existing challenges and open questions areas are also discussed.

8 Acknowledgements

This material is supported by the National Science Foundation (NSF) under grants 2006844 and the Cisco Faculty Research Award.

References

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Controlling popularity bias in learning-to-rank recommendation. In *RecSys*, 2017.

[2] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. The unfairness of popularity bias in recommendation. In *RecSys*, 2019.

[3] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. Towards a unified framework for fair and stable graph representation learning. *arXiv preprint arXiv:2102.13186*, 2021.

[4] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM*, 2006.

[5] Junaid Ali, Mahmoudreza Babaei, Abhijnan Chakraborty, Baharan Mirzasoleiman, Krishna P Gummadi, and Adish Singla. On the fairness of time-critical influence maximization in social networks. In *NeurIPS*, 2019.

[6] Sihem Amer-Yahia, Senjuti Basu Roy, Ashish Chawlat, Gautam Das, and Cong Yu. Group recommendation: Semantics and efficiency. *VLDB*, 2009.

[7] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*. 2016.

[8] Mario Arduini, Lorenzo Noci, Federico Pirovano, Ce Zhang, Yash Raj Shrestha, and Bibek Paudel. Adversarial learning for debiasing knowledge graph embeddings. In *SIGKDD*, 2020.

[9] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[10] James Atwood, Hansa Srinivasan, Yoni Halpern, and David Sculley. Fair treatment allocations in social networks. In *workshop of NeurIPS*, 2019.

[11] Abhijit Banerjee, Arun G Chandrasekhar, Esther Duflo, and Matthew O Jackson. The diffusion of microfinance. *Science*, 2013.

[12] Reuben Binns. Fairness in machine learning: Lessons from political philosophy. In *FAccT*, 2018.

[13] Reuben Binns. On the apparent conflict between individual and group fairness. In *FAccT*, 2020.

[14] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *NeurIPS*, 2016.

[15] Pietro Bongini, Monica Bianchini, and Franco Scarselli. Molecular generative graph neural networks for drug discovery. *Neurocomputing*, 2021.

[16] Ludovico Boratto, Gianni Fenu, and Mirko Marras. Interplay between upsampling and regularization for provider fairness in recommender systems. In *UMUAI*, 2020.

[17] Rodrigo Borges and Kostas Stefanidis. On measuring popularity bias in collaborative filtering data. In *workshop of EDBT/ICDT*, 2020.

[18] Avishek Bose and William Hamilton. Compositional fairness constraints for graph embeddings. In *ICML*, 2019.

[19] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 1998.

[20] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAccT*, 2018.

[21] Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo Ordoñez-Gauger. Balanced neighborhoods for fairness-aware collaborative recommendation. In *RecSys*, 2017.

[22] Kai Burkholder, Kenny Kwock, Yuesheng Xu, Jiaxin Liu, Chao Chen, and Sihong Xie. Certification and trade-off of multiple fairness criteria in graph-based spam detection. In *CIKM*, 2021.

[23] Jenna Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, 2016.

[24] Maarten Buyl and Tijl De Bie. The kl-divergence between a graph model and its fair i-projection as a fairness regularizer. In *ECML-PKDD*, 2021.

[25] Maarten Buyl and Tijl De Bie. Debayes: a bayesian method for debiasing network embeddings. In *ICML*, 2020.

[26] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 2017.

[27] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In *RecSys*, 2011.

[28] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *CSUR*, 2020.

[29] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *CIKM*, 2009.

[30] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240*, 2020.

[31] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, 2020.

[32] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *SIGKDD*, 2009.

[33] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. ESAM: discriminative domain adaptation with non-displayed items to improve long-tail performance. In *SIGIR*, 2020.

[34] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *NeurIPS*, 2018.

[35] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. In *NeurIPS*, 2019.

[36] Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *SIGKDD*, 2020.

[37] Sean Current, Yuntian He, Saket Gurukar, and Srinivasan Parthasarathy. Fairmod: Fair link prediction and recommendation via graph modification. *arXiv preprint arXiv:2201.11596*, 2022.

[38] Enyan Dai and Suhang Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM*, 2021.

[39] Yushun Dong, Jian Kang, Hanghang Tong, and Jundong Li. Individual fairness for graph neural networks: A ranking based approach. In *SIGKDD*, 2021.

[40] Yushun Dong, Ninghao Liu, Brian Jalaian, and Jundong Li. EDITS: modeling and mitigating data bias for graph neural networks. In *WWW*, 2022.

[41] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems*, 2020.

[42] Xin Du, Yulong Pei, Wouter Duivesteijn, and Mykola Pechenizkiy. Fairness in network representation by latent structural heterogeneity in observational data. In *AAAI*, 2020.

[43] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science*, 2012.

[44] Wei Fan, Kunpeng Liu, Rui Xie, Hao Liu, Hui Xiong, and Yanjie Fu. Fair graph auto-encoder for unbiased graph representations with wasserstein distance. In *ICDM*, 2021.

[45] Golnoosh Farnad, Behrouz Babaki, and Michel Gendreau. A unifying framework for fairness-aware influence maximization. In *Companion Proceedings of WWW*, 2020.

[46] Golnoosh Farnadi, Pigi Kouki, Spencer K. Thompson, Sriram Srinivasan, and Lise Getoor. A fairness-aware hybrid recommender system. In *workshop of RecSys*, 2018.

[47] Benjamin Fish, Ashkan Bashardoust, Danah Boyd, Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. Gaps in information access in social networks? In *WWW*, 2019.

[48] Joseph Fisher, Arpit Mittal, Dave Palfrey, and Christos Christodoulopoulos. Debiasing knowledge graph embeddings. In *EMNLP*, 2020.

[49] Joseph Fisher, Dave Palfrey, Christos Christodoulopoulos, and Arpit Mittal. Measuring social bias in knowledge graph embeddings. In *workshop of AKBC*, 2020.

[50] Danilo Franco, Nicolò Navarin, Michele Donini, Davide Anguita, and Luca Oneto. Deep fair models for complex data: Graphs labeling and explainable face recognition. *Neurocomputing*, 2022.

[51] Cornelius Fritz, Emilio Dorigatti, and David Rügamer. Combining graph neural networks and spatio-temporal disease models to predict covid-19 cases in germany. *Scientific Reports*, 2021.

[52] Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, et al. Fairness-aware explainable recommendation over knowledge graphs. In *SIGIR*, 2020.

[53] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *PNAS*, 2018.

[54] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, and Yongfeng Zhang. Towards long-term fairness in recommendation. In *WSDM*, 2021.

[55] Yingqiang Ge, Xiaoting Zhao, Lucia Yu, Saurabh Paul, Diane Hu, Chu-Cheng Hsieh, and Yongfeng Zhang. Toward pareto efficient fairness-utility trade-off inrecommendation through reinforcement learning. In *WSDM*, 2022.

[56] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *SIGKDD*, 2016.

[57] Edward L Grubb and Gregg Hupp. Perception of self, generalized stereotypes, and brand selection. *Journal of Marketing research*, 1968.

[58] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *IJCAI*, 2013.

[59] Shubham Gupta and Ambedkar Dukkipati. Protecting individual interests across clusters: Spectral clustering with guarantees. *arXiv preprint arXiv:2105.03714*, 2021.

[60] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[61] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *NeurIPS*, 2016.

[62] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *TIIS*, 2015.

[63] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, 2016.

[64] Kalervo J. and Jaana K. Cumulated gain-based evaluation of ir techniques. *TOIS*, 2002.

[65] Zeinab S Jalali, Weixiang Wang, Myunghwan Kim, Hema Raghavan, and Sucheta Soundarajan. On the information unfairness of social networks. In *SDM*, 2020.

[66] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *arXiv preprint arXiv:2101.11174*, 2021.

[67] Zhimeng Jiang, Xiaotian Han, Chao Fan, Zirui Liu, Na Zou, Ali Mostafavi, and Xia Hu. Fmp: Toward fair graph message passing against topology bias. *arXiv preprint arXiv:2202.04187*, 2022.

[68] Zhimeng Jiang, Xiaotian Han, Chao Fan, Fan Yang, Ali Mostafavi, and Xia Hu. Generalized demographic parity for group fairness. In *ICLR*, 2021.

[69] Kareem L Jordan and Tina L Freiburger. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. *J Ethn Crim Justice*, 2015.

[70] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *KAIS*, 2012.

[71] Toshihiro Kamishima and Shotaro Akaho. Considerations on recommendation independence for a find-good-items task. In *RecSys*, 2017.

[72] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Efficiency improvement of neutrality-enhanced recommendation. In *RecSys*, 2013.

[73] Jian Kang, Jingrui He, Ross Maciejewski, and Hanghang Tong. Inform: Individual fairness on graph mining. In *SIGKDD*, 2020.

[74] Jian Kang and Hanghang Tong. Fair graph mining. In *CIKM*, 2021.

[75] Jian Kang, Yan Zhu, Jiebo Luo, Yinglong Xia, and Hanghang Tong. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. In *WWW*, 2022.

[76] Mesut Kaya, Derek G. Bridge, and Nava Tintarev. Ensuring fairness in group recommendations by rank-sensitive balancing of relevance. In *RecSys*, 2020.

[77] Ahmad Khajehnejad, Moein Khajehnejad, Mahmoudreza Babaei, Krishna P Gummadi, Adrian Weller, and Baharan Mirzasoleiman. Crosswalk: Fairness-enhanced node representation learning. In *AAAI*, 2022.

[78] Moein Khajehnejad, Ahmad Asgharian Rezaei, Mahmoudreza Babaei, Jessica Hoffmann, Mahdi Jalili, and Adrian Weller. Adversarial graph embeddings for fair influence maximization over social networks. In *IJCAI*, 2020.

[79] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[80] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. In *ICML*, 2019.

[81] Öykü Deniz Köse and Yanning Shen. Fairness-aware node representation learning. *arXiv preprint arXiv:2106.05391*, 2021.

[82] Öykü Deniz Köse and Yanning Shen. Fair node representation learning via adaptive data augmentation. *arXiv preprint arXiv:2201.08549*, 2022.

[83] Emmanouil Krasanakis, Symeon Papadopoulos, and Ioannis Kompatsiaris. Applying fairness constraints on graph node ranks under personalization bias. In *CNA*, 2020.

[84] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. An adversarial approach to improve long-tail performance in neural collaborative filtering. In *CIKM*, 2018.

[85] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *NeurIPS*, 2017.

[86] Jurek L., Avishek A., and Megha K. User fairness in recommender systems. In *Companion Proceedings of WWW*, 2018.

[87] Charlotte Laclau, Ievgen Redko, Manvi Choudhary, and Christine Largeron. All of the fairness for edge prediction with optimal transport. In *AISTATS*, 2021.

[88] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. Operationalizing individual fairness with pairwise fair representations. *VLDB*, 2019.

[89] Anja Lambrecht and Catherine Tucker. Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads. *Management science*, 2019.

[90] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *TWEB*, 2007.

[91] Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. On dyadic fairness: Exploring and mitigating bias in graph connections. In *ICLR*, 2020.

[92] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *ICLR*, 2018.

[93] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. User-oriented fairness in recommendation. In *WWW*, 2021.

[94] Zhao Li, Xin Shen, Yuhang Jiao, Xuming Pan, Pengcheng Zou, Xianling Meng, Chengwei Yao, and Jiajun Bu. Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. In *ICDE*, 2020.

[95] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *JASIST*, 2007.

[96] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *SIGKDD*, 2010.

[97] Xiao Lin, Min Zhang, Yongfeng Zhang, Zhaoquan Gu, Yiqun Liu, and Shaoping Ma. Fairness-aware group recommendation with pareto-efficiency. In *RecSys*, 2017.

[98] Haifeng Liu, Nan Zhao, Xiaokun Zhang, Hongfei Lin, Liang Yang, Bo Xu, Yuan Lin, and Wenqi Fan. Dual constraints and adversarial learning for fair recommenders. *Knowledge-Based Systems*, 2022.

[99] Weiwen Liu and Robin Burke. Personalizing fairness-aware reranking. *arXiv preprint arXiv:1809.02921*, 2018.

[100] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A*, 2011.

[101] Farzan M., Tyler W., Heng Y., P. Tan, and A. Esfahanian. Bursting the filter bubble: Fairness-aware network link prediction. In *AAAI*, 2020.

[102] Ninareh M., Fred M., Nripsuta S., Kristina L., and Aram G. A survey on bias and fairness in machine learning. *CSUR*, 2021.

[103] Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. Subgroup generalization and fairness of graph neural networks. In *NeurIPS*, 2021.

[104] Jing Ma, Ruocheng Guo, Mengting Wan, Longqi Yang, Aidong Zhang, and Jundong Li. Learning fair node representations with graph counterfactual fairness. In *WSDM*, 2022.

[105] Ladislav Malecek and Ladislav Peska. Fairness-preserving group recommendations with user weighting. In *Adjunct Publication of UMAP*, 2021.

[106] Natalia Martinez, Martin Bertran, and Guillermo Sapiro. Fairness with minimal harm: A pareto-optimal approach for healthcare. *arXiv preprint arXiv:1911.06935*, 2019.

[107] Julian J McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *NeurIPS*, 2012.

[108] Ninareh Mehrabi, Muhammad Naveed, Fred Morstatter, and Aram Galstyan. Exacerbating algorithmic bias through fairness attacks. In *AAAI*, 2021.

[109] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Algorithmic fairness: Choices, assumptions, and definitions. *Annu. Rev. Stat. Appl.*, 2021.

[110] Nicolo N., Luca O., and Michele D. Learning deep fair graph neural networks. In *ESANN*, 2020.

[111] Mark EJ Newman. Modularity and community structure in networks. *PNAS*, 2006.

[112] Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Hongbo Deng, and Zhenzhong Chen. A dual heterogeneous graph attention network to improve long-tail performance for shop search in e-commerce. In *SIGKDD*, 2020.

[113] John Palowitch and Bryan Perozzi. Debiasing graph representations via metadata-orthogonal training. 2020.

[114] John Palowitch and Bryan Perozzi. MONET: debiasing graph embeddings via the metadata-orthogonal training unit. In *ASONAM*, 2020.

[115] Marlotte Pannekoek and Giacomo Spigler. Investigating trade-offs in utility, fairness and differential privacy in neural networks. *arXiv preprint arXiv:2102.05975*, 2021.

[116] Eli Pariser. The filter bubble: What the internet is hiding. *Policy Perspectives*, 2012.

[117] Gourab K. Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *WWW*, 2020.

[118] Judea Pearl. *Causality*. Cambridge university press, 2009.

[119] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, 2014.

[120] Dana Pessach and Erez Shmueli. Algorithmic fairness. *arXiv preprint arXiv:2001.09784*, 2020.

[121] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: An overview. *VLDB*, 2021.

[122] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. In *IJCAI*, 2019.

[123] Aida Rahmattalabi, Shahin Jabbari, Himabindu Lakkaraju, Phebe Vayanos, Max Izenberg, Ryan Brown, Eric Rice, and Milind Tambe. Fair influence maximization: A welfare optimization approach. In *AAAI*, 2021.

[124] Aida Rahmattalabi, Phebe Vayanos, Anthony Fulginiti, Eric Rice, Bryan Wilder, Amulya Yadav, and Milind Tambe. Exploring algorithmic fairness in robust graph covering problems. In *NeurIPS*, 2019.

[125] John Rawls. *A theory of justice: Revised edition*. 1999.

[126] Veronica Red, Eric D Kelsic, Peter J Mucha, and Mason A Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM review*, 2011.

[127] Soumajyoti Sarkar and Hamidreza Alvari. Mitigating bias in online microfinance platforms: A case study on kiva. org. In *ECML-PKDD*, 2020.

[128] Akrati Saxena, George Fletcher, and Mykola Pechenizkiy. Hmeiict: Fairness-aware link prediction in complex networks using community information. *J Comb Optim*, 2021.

[129] Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means. In *workshop of WAOA*, 2019.

[130] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 2008.

[131] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *CVPR*, 2018.

[132] Chiappa Silvia, Jiang Ray, Stepleton Tom, Pacchiano Aldo, Jiang Heinrich, and Aslanides John. A general approach to fairness with optimal transport. In *AAAI*, 2020.

[133] Edward H Simpson. Measurement of diversity. *nature*, 1949.

[134] Michael Skirpan and Micha Gorelick. The authority of "fair" in machine learning. In *workshop of FAccT*, 2017.

[135] David Solans, Battista Biggio, and Carlos Castillo. Poisoning attacks on algorithmic fairness. In *ECML-PKDD*, 2020.

[136] Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. Biased edge dropout for enhancing fairness in graph representation learning. *TAI*, 2021.

[137] Ana-Andreea Stoica and Augustin Chaintreau. Fairness in social influence maximization. In *Companion Proceedings of WWW*, 2019.

[138] Ana-Andreea Stoica, Jessy Xinyi Han, and Augustin Chaintreau. Seeding network influence in biased networks and the benefits of diversity. In *WWW*, 2020.

[139] Ana-Andreea Stoica, Christopher Riederer, and Augustin Chaintreau. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. In *WWW*, 2018.

[140] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2012.

[141] Latanya Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 2013.

[142] Lubos Takac and Michal Zabovsky. Data analysis in public social networks. In *DTI*, 2012.

[143] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *SIGKDD*, 2008.

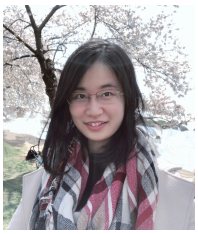
[144] Jiliang Tang, Huiji Gao, and Huan Liu. mtrust: Discerning multi-faceted trust in a connected world. In *WSDM*, 2012.

[145] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Investigating and mitigating degree-related biases in graph convoltuional networks. In *CIKM*, 2020.

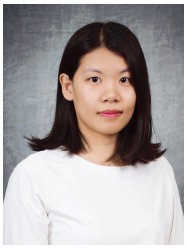
[146] Ya-Wen Teng, Hsi-Wen Chen, De-Nian Yang, Yvonne-Anne Pignolet, Ting-Wei Li, and Lydia Chen. On influencing the influential: disparity seeding. In *CIKM*, 2021.

[147] JH Trowbridge. On a technique for measurement of turbulent shear stress in the presence of surface waves. *JTECH*, 1998.
[148] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. Group-fairness in influence maximization. In *IJCAI*, 2019.
[149] Sotiris Tsioutsiouliklis, Evaggelia Pitoura, Panayiotis Tsaparas, Ilias Kleftakis, and Nikos Mamoulis. Fairness-aware pagerank. In *WWW*, 2021.
[150] Martijn PA van den Ende and J-P Ampuero. Automated seismic source characterization using deep graph neural networks. *Geophysical Research Letters*, 2020.
[151] Lav R Varshney, Nitish Shirish Keskar, and Richard Socher. Pretrained ai models: Performativity, mobility, and change. *arXiv preprint arXiv:1909.03290*, 2019.
[152] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.
[153] Blake W., Suriya G., Mesrob I O., and Nathan S. Learning non-discriminatory predictors. In *COLT*, 2017.
[154] Mengting Wan, Jianmo Ni, Rishabh Misra, and Julian J. McAuley. Addressing marketing bias in product recommendations. In *WSDM*, 2020.
[155] Daixin Wang, Jianbin Lin, Peng Cui, Quanhui Jia, Zhen Wang, Yanming Fang, Quan Yu, Jun Zhou, Shuang Yang, and Yuan Qi. A semi-supervised graph attentive network for financial fraud detection. In *ICDM*, 2019.
[156] Nan Wang, Lu Lin, Jundong Li, and Hongning Wang. Unbiased graph embedding with biased graph observations. In *WWW*, 2022.
[157] Jacek Wasilewski and Neil Hurley. Incorporating diversity in a learning to rank recommender system. In *FLAIRS*, 2016.
[158] Pak-Hang Wong. Democratizing algorithmic fairness. *Philosophy & Technology*, 2020.
[159] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. Learning fair representations for recommendation: A graph-based perspective. In *WWW*, 2021.
[160] Tiankai Xie, Yuxin Ma, Jian Kang, Hanghang Tong, and Ross Maciejewski. Fairrankvis: A visual analytics framework for exploring algorithmic fairness in graph mining models. *TVCG*, 2021.
[161] Xiaoying Xing, Hongfu Liu, Chen Chen, and Jundong Li. Fairness-aware unsupervised feature selection. In *CIKM*, 2021.
[162] Jiacheng Xiong, Zhaoping Xiong, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Graph neural networks for automated de novo drug design. *Drug Discovery Today*, 2021.
[163] B. Xu, H. Shen, B. Sun, et al. Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field. In *AAAI*, 2021.
[164] Bingke Xu, Yue Cui, Zipeng Sun, Liwei Deng, and Kai Zheng. Fair representation learning in knowledge-aware recommendation. In *ICBK*, 2021.
[165] K. Xu, W. Hu, Jure L., and Stefanie J. How powerful are graph neural networks? In *ICLR*, 2019.
[166] Amulya Yadav, Bryan Wilder, Eric Rice, Robin Petering, Jaih Craddock, Amanda Yoshioka-Maxwell, Mary Hemler, Laura Onasch-Vera, Milind Tambe, and Darlene Woo. Bridging the gap between theory and practice in influence maximization: Raising awareness about hiv among homeless youth. In *IJCAI*, 2018.
[167] Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. In *NIPS*, 2017.
[168] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Syst. Appl.*, 2009.
[169] Karen Yeung. Algorithmic regulation: A critical interrogation. *Regulation & Governance*, 2018.
[170] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *ICML*, 2013.
[171] Ziqian Zeng, Rashidul Islam, et al. Fair representation learning for heterogeneous information networks. In *AAAI*, 2021.
[172] Wenbin Zhang, Jeremy C Weiss, Shuigeng Zhou, and Toby Walsh. Fairness amidst non-iid graph data: A literature review. *arXiv preprint arXiv:2202.07170*, 2022.
[173] Xu Zhang, Liang Zhang, Bo Jin, and Xinjiang Lu. A multi-view confidence-calibrated framework for fair and stable graph representation learning. In *ICDM*, 2021.
[174] Ziwei Zhu, Yun He, et al. Popularity-opportunity bias in collaborative filtering. In *WSDM*, 2021.
[175] Ziwei Zhu, Jianling Wang, and James Caverlee. Measuring and mitigating item under-recommendation bias in personalized ranking systems. In *SIGIR*, 2020.

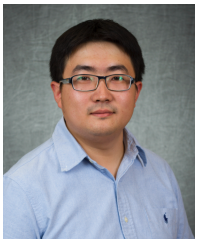
Yushun Dong Yushun Dong is a Ph.D. student in the Department of Electrical and Computer Engineering at the University of Virginia. He received the B.S. degree in Telecommunications from Beijing University of Posts and Telecommunications in 2019. His research interests are broadly in data mining and machine learning, with a particular focus on graph learning algorithms. In the past few years, his works have been published in top-tier venues including SIGKDD, WWW, and CIKM.



Jing Ma Jing Ma is a Ph.D. candidate in the Department of Computer Science at the University of Virginia. Her research interests include causal inference, machine learning, data mining, and especially for bridging the gap between causality and machine learning. Her works have been published in top conferences and journals such as IJCAI, WWW, AAAI, TKDE, WSDM, SIGIR, and IPSN.



Chen Chen Chen Chen is currently a Research Assistant Professor at the University of Virginia. Before joining the University of Virginia, Chen was a software engineer at Google working on personalized recommendations for Google Assistant. Chen got her Ph.D. degree from Arizona State University. Her research focuses on the connectivity of complex networks and has been applied to address pressing challenges in various high-impact domains, including social media, bioinformatics, recommendation, and critical infrastructure systems. Her research has appeared in top-tier conferences (including KDD, ICDM, SDM, WSDM, SIGIR, AAAI, and IJCAI), and prestigious journals (including IEEE TKDE, ACM TKDD, and SIAM SAM). Chen has received several awards, including Bests of SDM'15, Bests of KDD'16, Rising Star in EECS'19, Outstanding Reviewer of WSDM'21.



Jundong Li Jundong Li is an Assistant Professor in the Department of Electrical and Computer Engineering, with a joint appointment in the Department of Computer Science, and the School of Data Science. He received Ph.D. degree in Computer Science at Arizona State University in 2019. His research interests are in data mining, machine learning, and causal inference. He has published over 100 articles in high-impact venues (e.g., KDD, WWW, AAAI, IJCAI, WSDM, EMNLP, CSUR, TPAMI, TKDE, TKDD, and TIST.). He has won several prestigious awards, including NSF CAREER Award, JP Morgan Chase Faculty Research Award, Cisco Faculty Research Award, and being selected for the AAAI 2021 New Faculty Highlights program.