FairLP: Towards Fair Link Prediction on Social Network Graphs

Yanying Li, Xiuling Wang, Yue Ning, Hui Wang
Stevens Institute of Technology
{yli158, xwang193, hwang4, yue.ning}@stevens.edu

Abstract

Link prediction has been widely applied in social network analysis. Despite its importance, link prediction algorithms can be biased by disfavoring the links between individuals in particular demographic groups. In this paper, we study one particular type of bias, namely, the bias in predicting *inter-group* links (i.e., links across different demographic groups). First, we formalize the definition of bias in link prediction by providing quantitative measurements of *accuracy disparity*, which measures the difference in prediction accuracy of inter-group and intra-group links. Second, we unveil the existence of bias in six existing state-of-the-art link prediction algorithms through extensive empirical studies over real-world datasets. Third, we identify the imbalanced density across intra-group and inter-group links in training graphs as one of the underlying causes of bias in link prediction. Based on the identified cause, fourth, we design a pre-processing bias mitigation method named FAIRLP to modify the training graph, aiming to balance the distribution of intra-group and inter-group links while preserving the network characteristics of the graph. FAIRLP is model-agnostic and thus is compatible with any existing link prediction algorithm. Our experimental results on real-world social network graphs demonstrate that FAIRLP achieves better trade-off between fairness and prediction accuracy than the existing fairness-enhancing link prediction methods.

Introduction

Link prediction is an important task for network analysis. It studies interactions among individuals and infers new relations that may appear in the future of evolving networks. Given the ubiquitous existence of social networks, link prediction has been used widely in recommendations on social networks (Adamic and Adar 2003; Guy and Pizzato 2016; Sanz-Cruzado and Castells 2019).

Although social recommendations such as friend suggestions and *who-to-follow* have become increasingly popular and influential on the growth of social media, potential algorithmic bias in recommendation systems may lead to negative effects towards some minority groups in social networks. For example, biased recommendations can lead to the *glass ceiling* effect[1] against female users (Stoica and Chaintreau 2018) and the *rich-get-richer effect* against minority groups (Masrour et al. 2020; Fabbri et al. 2020). In particular, due to the homophily principle (McPherson, Smith-Lovin, and Cook 2001), social recommendation algorithms that capture such principle intend to promote links between pairs of individuals belonging to the same demographic group specified by particular features (e.g., gender and race) (Masrour et al. 2020). Such biased link predictions can lead to unintended significant consequences in many application domains that utilize users' social network data in automatic decision making. A typical example is the online peer-to-peer (P2P) financial lending platforms (e.g., LendingClub and Prosper) that utilize the lenders' social networks to evaluate their credibility. On these platforms, each borrower is associated with a *social credit score*. A borrower who is connected with more friends that have low loan default risk receives a high social credit score (Wei et al. 2016). These social credit scores are utilized by machine learning algorithms to determine if the borrower's loan application is to be approved or declined (Niu, Ren, and Li 2019; Freedman and Jin 2008). Intuitively, for those borrowers whose friends are dominantly of high default risk, a biased link prediction algorithm tends to recommend similar users of high loan default risk. Consequently the loan applications of those borrowers will be more likely to be denied in the future. This is in particular unfair to those borrowers who are themselves creditworthy but lack creditworthy social connections (Li et al. 2020b).

An important issue of investigating bias in social recommender systems is the measurement of bias. Algorithmic bias in recommendation algorithms over social networks have been measured in various metrics by prior work. For example, *disparity of visibility* in recommendations (Stoica and Chaintreau 2018; Fabbri et al. 2020) measures the difference in the number of times a particular user/group appears in the recommendations compared with other users/groups. An alternative measurement is to measure the change of network modularity (Masrour et al. 2020); the link prediction results are indeed biased if they create more inter-group (i.e., nodes belong to different groups) or intra-group links than

[1] The term "glass ceiling" refers to invisible barriers that keep some people from advancing in the workplace (Fabbri et al. 2020)

expected based on the null model for modularity. Bias also can be measured as the variance between the recommendation rates of different groups in recommendations (Rahman et al. 2019). Although these bias measurements are useful in interpreting the negative effects of algorithmic bias in recommendations towards social networks, none of them have considered an equally important metric, which measures the disparity in link prediction accuracy across different user groups. Indeed, a recommender system that achieves disparate link prediction accuracy on different groups may unfairly disadvantage some users (Masrour et al. 2020). For example, consider two groups G_1 and G_2 of P2P lending borrowers, where G_1 and G_2 contain users whose friends are dominantly of low and high default risk respectively. If a link prediction algorithm delivers high prediction accuracy to G_1 and low prediction accuracy to G_2 when recommending similar users to both groups, apparently the algorithm treats the users in G_2 unfairly as they are wrongly recommended with users of high default risk, which will greatly disadvantage their loan applications.

In this paper, *we focus on the analysis and mitigation of disparate prediction accuracy in link prediction for social recommender systems.* We assume the nodes are classified into two groups by a well-defined protected attribute (e.g., gender or race). Then we follow the prior work (Masrour et al. 2020) and classify the links into two groups, namely the *inter-group* links between nodes of the same group, and *intra-group* links between nodes that belong to different groups. We define bias as the disparity of link prediction accuracy (in terms of either positive rate or true positive rate) between two link groups. We study three important research questions that remain un-answered by the prior works:

- RQ_1: Do existing link prediction algorithms indeed have disparate accuracy across inter-group and intra-group links?
- RQ_2: If such accuracy disparity exists, what are its causes?
- RQ_3: How to fix the accuracy disparity while addressing the trade-off between fairness and prediction accuracy?

Contributions. To our best knowledge, our work presents the first comprehensive study of accuracy disparity in existing link prediction algorithms. Furthermore, while existing fairness-aware link prediction methods (Masrour et al. 2020; Rahman et al. 2019; Nilizadeh et al. 2016; Lee et al. 2019; Karimi et al. 2018) are limited to specific link prediction algorithms, we design the first bias mitigation algorithm that is compatible with most of existing link prediction algorithms. Our contributions are summarized as follows.

- We adapt two existing widely-used fairness notions, namely *statistical parity* and *equal opportunity*, to formalize the notion of accuracy disparity in link prediction. Accuracy disparity quantifies the difference in prediction accuracy of inter-group and intra-group edges.
- We measure the accuracy disparity of six state-of-the-art link prediction algorithms on three real-world social network graphs. Our empirical results show that accuracy disparity exist for all the tested prediction algorithms. In particular, the prediction accuracy of inter-group links is always lower than that of the intra-group links.
- We identify the imbalanced group density of two link groups as one of the underlying sources of accuracy disparity. We also identify network homophily as the cause of the imbalanced group density.
- We design a bias mitigation method named FAIRLP to remedy the imbalanced group density in the training graph by inserting/removing edges. FAIRLP mitigates the imbalanced group density across different groups, while minimizing the amounts of structure change on the original graph.
- We compare the performance of FAIRLP with three existing fairness-aware link prediction algorithms, and show that FAIRLP better addresses the trade-off between fairness and prediction accuracy than these methods.

Related Work

Fairness in machine learning. A multitude of formal, mathematical definitions of fairness in machine learning has been proposed in the last few years. These definitions can be categorized into two categories: (1) *Group fairness* that is concerned with the protected groups (such as racial or gender groups) and requires that some statistic of interest be approximately equalized across groups (Feldman et al. 2015; Calders and Verwer 2010; Hardt, Price, and Srebro 2016); and (2) *Individual fairness* (Dwork et al. 2012) that prevents discrimination against individuals and requires similar individuals are treated similarly. In this paper, we mainly focus on group fairness. In particular, we adapt two widely-used group fairness definitions – *statistical parity* (Feldman et al. 2015) and *equal opportunity* (Hardt, Price, and Srebro 2016) – to the graph learning setting.

Methodologically, the existing bias mitigation methods fall broadly into three categories: (1) *pre-processing*: the bias in the training data is mitigated (Calders, Kamiran, and Pechenizkiy 2009; Kamiran and Calders 2009; Feldman et al. 2015); (2) *in-processing*: the machine learning model is modified by adding fairness as additional constraint (Calders and Verwer 2010; Zafar et al. 2017; Goh et al. 2016); and (3) *post-processing*: the results of a previously trained classifier are modified to achieve the desired results on different groups (Hardt, Price, and Srebro 2016). Since the in-processing methods are not compatible with other existing link prediction algorithms, while the post-processing methods may degrade prediction accuracy significantly, we mainly focus on *pre-processing* disparity mitigation methods in this paper.

Bias in social recommender systems. Recent studies have shown the existence of various types of systematic bias in social recommender systems. (Lee et al. 2019) analyze the perception bias in social network, which depends on the level of homophily and its asymmetric nature, as well as the size of minority group. (Karimi et al. 2018) study the impact of homophily and group size on degree distribution and visibility in social networks, and observe homophily can put minority groups at a disadvantage when establishing links with a majority group. (Fabbri et al. 2020) also investigate the effect of homophily on visibility of minorities in people recommender systems, and find that homophily plays a key role in the disparate visibility of different groups. (Nilizadeh et al. 2016; Stoica and Chaintreau 2018) show that biased recommendations can bring the glass ceiling effect, which affects female groups negatively.

Fairness in network link prediction. (Kang et al. 2020) focus on individual fairness, which requires that similar users should receive similar link prediction results. Their fairness definition is fundamentally different from our group fairness definition. (Li et al. 2020a) consider *dyadic fairness* for link prediction, which requires the link prediction results is independent from the fact of whether two vertices at the link have the same sensitive attribute or not. (Rahman et al. 2019) considers the fairness of graph embedding, but with link prediction as one of the downstream learning tasks. In particular, they measure the bias in recommendations as the difference between the recommendation rate (positive rate) of different groups. They design a fairness-enhancing graph embedding algorithm named *Fairwalk* that adds the fairness constraint to node2vec (Grover and Leskovec 2016), a state-of-the-art node embedding algorithm. (Masrour et al. 2020) uses network modularity as the bias measurement, and defines the fairness goal as reducing network modularity by the link prediction results. This is different from our goal that requires parity in prediction accuracy across different groups. Furthermore, as the fairness-aware link prediction algorithm (named FLIP) designed by (Masrour et al. 2020) is an in-processing mitigation, it cannot be applied to any existing link prediction algorithm. On the contrary, our FairLP algorithm, which is a *pre-processing* bias mitigation method, is compatible with any existing link prediction algorithm.

Defining Fairness in Link Prediction

Conventional Group Fairness Definitions In this paper, we mainly focus on group fairness. In general, the group fairness model is defined as following: given a dataset consisting of n i.i.d. samples $\{(A_i, X_i, Y_i)\}_{i=1}^n$ with domain $A \times X \times Y$, where A denotes the *protected features* such as gender and race, X denotes the non-protected features, and Y is an outcome feature, a machine learning system ensures that the prediction model learned from these samples does not have discriminatory effects towards the *protected groups* defined by the values associated with the *protected attributes*. For simplicity, we only consider one protected group. In the following discussions, we use $A = 0$ and $A = 1$ to indicate the protected and un-protected groups respectively.

As fairness is a complex and multi-faceted concept which depends on many factors (e.g., context and domains), many statistical definitions of fairness have been introduced in the literature. A recent survey (Mehrabi et al. 2019) summarizes over ten fairness definitions that are widely used in the literature: each is relevant to specific scenarios and data types, and yet none is universally applicable. In this paper, we consider two widely-used fairness definitions - *statistical parity* and *equal opportunity*, explained below.

Statistical parity. Also known as demographic parity, statistical parity is one of the most intuitive and widely-used group fairness notions (Kang et al. 2021). It requires an equal probability of being classified with the positive label across different groups:

$$Pr(\hat{Y} = 1|A = 1) = Pr(\hat{Y} = 1|A = 0).$$

We consider statistical parity as it allows us to measure fairness independent of the ground truth (existing links) which in our case may be biased itself.

Equal opportunity (Hardt, Price, and Srebro 2016) considers true positive rates of protected and un-protected groups. Specifically, an algorithm is considered to be fair under equal opportunity if its true positive rates are the same across different groups. Formally,

$$Pr(\hat{Y} = 1|Y = 1, A = 1) = Pr(\hat{Y} = 1|Y = 1, A = 0).$$

Group Fairness in Link Prediction Next, we discuss how we adapt the notion of equal opportunity to link prediction. As the group fairness relies on the definition of protected groups, we first define both *protected node groups* and *protected edge groups*.

Protected node groups. We extend the definition of protected groups to the graph setting. Formally, consider the social network graph $\mathbf{G}(V, E)$ where each vertex $v \in V$ represents an individual, and each edge $(v, v') \in E$ represents a link between two individuals denoted by v and v'. Each node v is associated with a set of features $\mathcal{F}$ describing the individual's personal information such as age, gender, and race. We categorize the features into two types: *non-protected node features* X and *protected node features* A ($X \cup A = \mathcal{F}$). The examples of protected node features include demographic features such as race and gender. The *protected node group*, denoted as $V^\star$, is specified by adding value-based constraints on A (e.g., *gender* = "female" or *age* $<$ 18). All the remaining nodes that do not belong to $V^\star$ are grouped as the un-protected node group (denoted as $\overline{V^\star}$). For simplicity, we only consider one protected node feature and one protected group in this paper. Our results can be easily extended to multiple protected node features as well as multiple protected/un-protected groups.

Consider an example of social network graph in which each node is associated with a *gender* feature, which is specified as the protected node feature. If the male users dominate the whole population in the graph, the nodes of male users (i.e., *gender = "M"*) construct the un-protected node group, while the nodes of female users (i.e., *gender = "F"*) are considered as the protected node group.

Protected edge groups. Given a set of edges E in graph $\mathbf{G}$, an edge group $E' \subseteq E$ is defined by the node feature $f \in \mathcal{F}$ of E':

$$E' = \{e(v, v')|v.f = a \text{ and } v'.f = a'\},$$

where $v.f$ indicates the associated value of node v on the feature f, and a and a' can be the same. We call this edge group the *(a-a') edge group*. In the running example, there are three edge groups on the gender feature: (M-M), (F-F), and (M-F). In general, given a protected node feature that has ℓ distinct values, there are $\frac{(\ell+1)\ell}{2}$ edge groups.

Among all the edge groups, we define the *protected edge group* (denoted as $E^\star$) as the union of all edge groups that connect nodes across the protected and un-protected node groups. Formally,

$$E^\star = \{e(v, v')|v \in V^\star \text{ and } v' \in \overline{V^\star}\}.$$

The remaining links are included in a group called the *unprotected edge group* (denoted as $\overline{E^\star}$). In this paper, we only consider binary grouping memberships (i.e., one protected and one unprotected edge group, where the inter-group links are considered as the protected edge group because the inter-group links between different demographic groups are more likely to be demoted than the intra-group links by the link prediction algorithms due to the homophily principle (McPherson, Smith-Lovin, and Cook 2001) (i.e., individuals tend to form social ties with other similar individuals in a network). We leave fairness issues of within-group link prediction and non-binary group memberships in our future work.

In the running example, as *gender = "F"* is defined as the protected node group, the protected edge group $E^\star$ includes all inter-gender edges (i.e., M-F edges), while the un-protected edge group $\overline{E^\star}$ includes all intra-gender edges (i.e., (M-M) and (F-F) edges).

Accuracy disparity. In this paper, we adapt both statistical parity and equal opportunity notions to link prediction. we use $Y = 1$ to denote the observed edges in a ground truth graph and $\hat{Y} = 1$ denotes the predicted edges. Formally, given a graph $\mathbf{G}(V, E)$ and its predicted graph $\hat{\mathbf{G}}(V, \hat{E})$, we measure two types of *accuracy disparity* that measures the disparity of accuracy across different groups:

- *Positive rate disparity* (**PD**) is adapted from the notion of statistical parity:
$$\text{PD} = Pr(\hat{Y} = 1|E^\star) - Pr(\hat{Y} = 1|\overline{E^\star}). \quad (1)$$
- *True-positive rate disparity* (**TPD**) is adapted from equal opportunity:
$$\text{TPD} = Pr(\hat{Y} = 1|Y = 1, E^\star) - Pr(\hat{Y} = 1|Y = 1, \overline{E^\star}). \quad (2)$$

Both PD and TPD measurements are equally important and complementary in terms of fairness in link prediction. Intuitively, PD measures the difference between the positive rate of two different edge groups, while TPD measures the difference between the true positive rate for these two groups. Negative PD values from a link prediction algorithm indicate that this algorithm promotes more edges in unprotected edge groups (e.g., M-M and F-F groups) than protected groups (e.g., M-F group), while negative TPD values indicate that a prediction algorithm wrongly promotes more edges from unprotected groups than the protected one. We are aware the existence of other accuracy measurements (e.g., false positive rate). These accuracy measurements will be left to the future work.

Accuracy Disparity of Link Prediction

To answer the research question RQ_1 - *Is unfairness in link prediction a real problem?*, we perform a set of empirical studies to evaluate the accuracy disparity of several state-of-the-art link prediction algorithms. In this section, we present our evaluation results. All experiments are executed on a machine with 2×Intel(R) Xeon(R) 12-core CPUs and 128 GB memory. All algorithms are implemented in Python. **Datasets.** We use three real-world datasets, namely Google+, Facebook, and DBLP datasets, in our experiments. We consider these three public datasets because they are popularly used in the literature (Masrour et al. 2020; Palowitch and Perozzi 2019; Kang et al. 2020) for fairness analysis. Table 1 summarizes the main properties of these datasets. **Fairness setup.** In all three datasets, we consider *gender*, the only demographic feature that the three datasets include, as the protected node feature. We consider the nodes that take minority as the protected node group (Female nodes in both Google+ and DBLP datasets, and male nodes in Facebook dataset). The group of inter-gender links (i.e., M-F edges) is defined as the protected edge group. The un-protected edge group includes all F-F and M-M edges.

Table 1: Description of datasets. The numbers (x, y%) in the parentheses indicate the number of nodes/edges in the protected node/edge group and its percentage in the group.

	Google+	**Facebook**	**DBLP**
#nodes	4,417	1,050	10,000
#edges	119,582	24,191	37,430
Protected node group	Female (1,455, 33%)	Male (459, 44%)	Female (2,174, 22%)
Protected edge group	M-F (22,786, 19%)	M-F (6,294, 26%)	M-F (5,752, 15%)

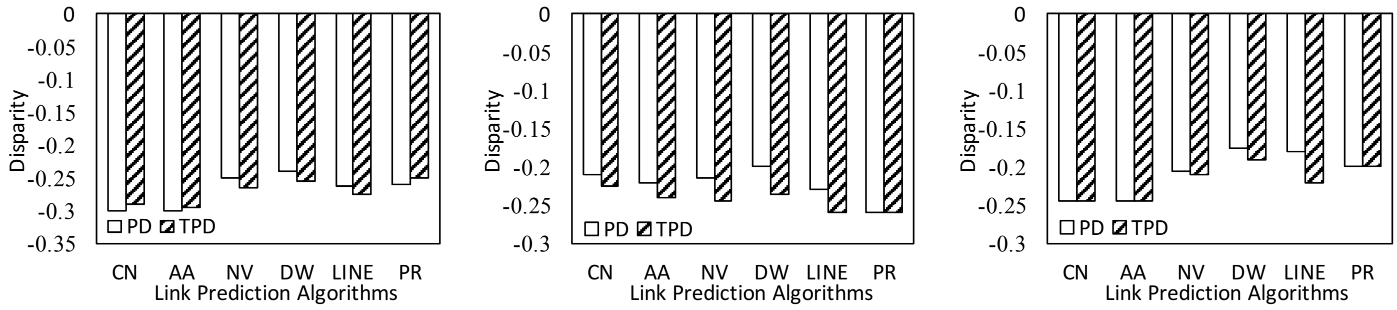
Link prediction algorithms. We consider two different types of link prediction algorithms, namely *similarity-based* algorithms and *graph embedding based* algorithms, and six state-of-the-art link prediction algorithms in these two categories:

- *Similarity-based* algorithms: We consider two conventional algorithms: common neighbors (**CN**) (Liben-Nowell and Kleinberg 2007) and Adamic-Adar index (**AA**) (Adamic and Adar 2003). Both predict the links between nodes based on the similarity of their neighbors.
- *Random walk based* algorithm: We consider the **PageRank** algorithm (Brin and Page 1998) that models the links between the seed node and any other node in the network via a Markov chain produced by random walk, and associates a PageRank score with each node. A high PageRank score indicates a possible link between the its associated node and the seed node.
- *Graph embedding based* algorithms: We consider link prediction that use three state-of-the-art graph embedding methods: **node2vec** (Grover and Leskovec 2016), **DeepWalk** (Perozzi, Al-Rfou, and Skiena 2014), and **Line** (Tang et al. 2015). These three algorithms convert nodes to low-dimensional vectors that preserve proximity. The links are predicted based on the node embedding.

In our experiments, we randomly select 60% of the edges for training, and 40% for testing.

Accuracy of Link Prediction Algorithms

We measure *area under the receiver operating characteristic curve* (AUC) as the link prediction accuracy. In general, all six algorithms deliver satisfying accuracy performance on the three graphs (AUC $\in [0.77, 0.97]$). We omit the details due to the limited space.



(a) Google+

(b) Facebook

(c) DBLP

Figure 1: Accuracy disparity of link prediction algorithms (NV: node2vec, DW: DeepWalk, PR: PageRank)

Evaluation of Accuracy Disparity

Although all the prediction algorithms perform well over the whole graph, do they perform differently for different edge groups? To answer this question, next, we measure **TPD** disparity of the protected edge group, i.e., inter-gender edge group, by the six link prediction algorithms. The results are shown in Figure 1. We summarize the major observations as follows.

Non-negligible accuracy disparity exists. The most notable and important finding is that *accuracy disparity exists for all the examined link prediction algorithms.* Each prediction algorithm shows negative accuracy disparity towards the protected edge group, meaning they indeed discriminate against the protected edge group. For example, the PD and TPD disparities of Line algorithm on Facebook dataset are around -0.2 and -0.25 respectively, which indicates that while Line promotes more intra-gender links than inter-gender ones, such promotion is indeed incorrect and thus leads to the mistreatment of inter-gender edges by prediction. Indeed, the accuracy disparity across all settings is non-negligible. Specifically, PD and TPD disparity range in [-0.18, -0.3] and [-0.19, -0.3] in all settings.

Accuracy disparity varies across different algorithms and datasets. There is no absolute "winner" or "loser" (i.e., the algorithm that always delivers the largest/smallest PD and TPD) among the six prediction algorithms. In other words, none of these algorithms are inherently more biased than the others by design. Furthermore, the prediction algorithms show different amounts of accuracy disparity on different input graphs. For instance, the PD disparity of CN algorithm is -0.21 on Facebook dataset, and -0.24 on DBLP dataset. This implies that the accuracy disparity is related to the input training graph.

Causes of Accuracy Disparity

To answer the research question RQ_2: - *what causes unfairness in the existing link prediction algorithms*, in this section, we analyze the possible causes of the identified accuracy disparity.

Network Homophily

At a high level, network homophily refers to the tendency of individuals connecting with others similar to them (McPherson, Smith-Lovin, and Cook 2001). In this paper, we use *network modularity* (Newman and Girvan 2004) to quantify network homophily. Suppose a graph contains n nodes and these nodes are split into two groups. For each node v_i, let $c_i = 1$ if v_i belongs to group 1 and $c_i = -1$ if it belongs to group 2. In this paper, we consider protected and non-protected node groups. Let $A_{i,j}$ be the number of edges between nodes v_i and v_j (normally 0 or 1). Then the expected number of edges between node v_i and node v_j if edges are placed at random is $\frac{d_i d_j}{2m}$, where d_i and d_j are the degrees of v_i and v_j, and $m = \frac{1}{2}\sum_{v_i} d_i$ is the total number of edges in the graph. The network modularity O is measured as:

$$O = \frac{1}{2m}\sum_{v_i,v_j}(A_{i,j} - \frac{d_i d_j}{2m})c_i c_j. \tag{3}$$

Intuitively, when its value approaches 1 (maximum), it indicates that nodes in the same node group are more likely to be connected than the nodes from different groups.

Table 2: Network modularity

	Google+	**Facebook**	**DBLP**
Modularity	0.22	0.24	0.094
p(link)	0.012	0.044	0.00075
p(link|intra)	0.017	0.064	0.00096
p(link|inter)	0.0053	0.023	0.00034

To measure the effect of network homophily, we also measure three probabilities:

- $\mathbf{p}$(link): the probability of a pair of nodes being linked.
- $\mathbf{p}$(link|intra) and $\mathbf{p}$(link|inter): the conditional probability of a link existing between a pair of intra-group/inter-group nodes and inter-group nodes.

In Table 2, we show the measurement of network homophily and the three link probabilities of the three networks used in our experiments. All the three networks have the homophily property, where the modularity can be as large as 0.24. The link probabilities also demonstrate the existence of the homophily property, as the conditional probability of intra-group links is always much higher than that of the inter-groups links. Indeed, $\mathbf{p}$(link|intra) is around three times of $\mathbf{p}$(link|inter) on all the three graphs. In other words, the individuals are much more likely to connect with people of the same gender than those of different genders in these graphs.

Connection between Network Homophily and Accuracy Disparity

Prior research (Suresh and Guttag 2019; Tolan et al. 2019) has identified imbalanced data distribution across different groups is one of the underlying reasons for biased machine learning in general. In this section, we study which type(s) of imbalanced data distribution are the underlying causes of accuracy disparity, and what are the sources of such imbalanced data distribution.

Relationship between network homophily and imbalanced group density. The effect of network homophily leads to the difference in the two probability values $\mathbf{p}(\text{link}|\text{intra})$ and $\mathbf{p}(\text{link}|\text{inter})$. Indeed, these two values measure the *density* of the intra- and inter-group link groups. Informally, the group density measures the proportion of possible edges in the group that are indeed connected in the given graph. Higher density indicates that the nodes involved in the edge group are more strongly connected. Therefore, the difference between $\mathbf{p}(\text{link}|\text{intra})$ and $\mathbf{p}(\text{link}|\text{inter})$ measures the imbalance between density of different link groups. In other words, the imbalanced group density inherently exists in the network due to network homophily. **Relationship between imbalanced group density and accuracy disparity.** The relationship between network homophily and imbalanced group density does not directly imply that imbalanced group density causes accuracy disparity. Therefore, next, we quantitatively measure the correlation and causality between group density and accuracy disparity.

First, we measure the Pearson correlation between group density and prediction accuracy of each group, where prediction accuracy is measured in terms of both positive rate (PR) and true positive rate (TPR). It turned out that the Pearson correlation between group density and PR is 0.67, and the Pearson correlation between group density and TPR is 0.62. Such large Pearson correlation indicates that group density is strongly correlated with the prediction accuracy and thus the accuracy disparity.

To further examine whether imbalanced group density of different groups is the cause of accuracy disparity, then we measure pairwise causality between group density and PR/TPR by utilizing the popular Additive Noise Model (ANM) (Hoyer et al. 2008) provided by a causal discovery toolbox.[2] ANM model considers the bivariate case with two features x and y. It outputs a causation score that indicates the direction of causality between x and y (score = 1 if $x \rightarrow y$ and -1 if $y \rightarrow x$). The returned causation score are 0.96 for PR and 0.92 for TPR respectively. These high scores indicate that the group density is the cause of both PR and TPR. Therefore, disparate group density across protected and un-protected groups is one of the causes of the difference in PR/TPR across these groups, which leads to the accuracy disparity.

Disparity Mitigation

In this section, we answer the research question RQ_3 - *how to fix unfairness in link prediction.* As the imbalanced density across different edge groups is identified as one of the underlying causes of accuracy disparity, we design FAIRLP, a mitigation algorithm that removes this cause. FAIRLPis a pre-processing method that is compatible with any existing link prediction algorithm.

Problem Definition

Intuitively, FAIRLP aims to flatten the density of the protected and un-protected edge groups by adding/deleting edges from the given graph G. There exists the trade-off between fairness and model accuracy. Removing too many important edges, or adding too many new edges that do not follow the original graph characteristics, will affect the graph structure significantly, thus leading to substantial accuracy loss in link prediction. To address this trade-off issue, first, we quantify the importance of edges. Given an edge $e(v_i, v_j)$, we measure its weight $w(e)$ as following:

$$w(e) = |\Gamma(v_i) \cap \Gamma(v_j)|, \tag{4}$$

where $\Gamma(v)$ returns the 1-hop neighborhood of the node v.

Intuitively, the edge weight scheme respects the principle of many link prediction algorithms for social graphs – individuals who have more common friends should be more likely to be connected. Thus the edges between these nodes have higher importance.

We are aware of alternative edge weight schemes (e.g., based on number of common neighbors in the neighborhood up to $k > 1$ hops, or based on degrees of connecting nodes instead of their neighborhood). We will compare the performance of our neighborhood-based scheme with some of these alternative schemes in the experiments.

Based on the definition of node importance, we define the objective function of FAIRLP. Given a graph $\mathbf{G}(V, E)$, let $\mathbf{G}'(V', E')$ be the graph after applying FAIRLP on $\mathbf{G}$. Since FAIRLP only adds and removes edges, the nodes in both $\mathbf{G}$ and $\mathbf{G}'$ are the same (i.e., $V = V'$). FAIRLP aims to find a set of edges $E^\diamond = E^+ \cup E^-$, where $E^- \subseteq E$ refers to the edges to be deleted and $E^+ \subseteq (V \times V - E)$ refers to the edges to be added ($V \times V$ denotes all possible edges in $\mathbf{G}$), such that the difference between the sum of weights of all edges in $\mathbf{G}$ and $\mathbf{G}'$ is minimized. Clearly, $E' = E - E^- + E^+$. Formally,

$$\begin{aligned} \min_{E^-, E^+} \quad & |\sum_{e \in E} w(e) - \sum_{e \in (E - E^- + E^+)} w(e)| \\ \text{s.t.} \quad & density(E^*) = density(\overline{E^*}), \end{aligned} \tag{5}$$

where E^* and $\overline{E^*}$ are protected and non-protected edge groups, and density() returns the density of an edge group. Intuitively, FAIRLP tries to minimize the impact on the graph structure by minimizing the change on the total weights of all edges. We require equal density across different groups as a constraint as the disparity of group density has been shown as one of the causes of accuracy disparity.

Algorithm Details

Exactly optimizing Formula (5) is computationally infeasible, as solving it would require enumerating all possible density values $density(E^*)$ (and $density(\overline{E^*})$), which can be

[2] https://fentechsolutions.github.io/CausalDiscoveryToolbox/html/causality.html

Algorithm 1 FairLP

```
Calculate the density of protected and un-protected edge groups in $G_{train}$;
Calculate the flattened density $r$ ($r$ is calculated in different ways for AvgD, MinD, and MaxD);
for Protected/un-protected group EG do
    $c_{\text{max}}$: number of all possible edges in EG (Equation 6);
    $c_{\text{real}}$: number of edges in EG;
    $c_{\text{exp}} = c_{\text{max}} \cdot r$;    //Expected number of edges
    $k = c_{\text{exp}} - c_{\text{real}}$;
    Initiate $U_{\text{EG}}$ as the set of all possible edges in EG;
    Initiate EG contains all edges in EG, and NEG $= U_{\text{EG}} -$ EG;
    $count = 0$;
    if $k > 0$ then
        while $count <= k$ do
            Pick the edge $e \in$ NEG of the lowest weight;
            Add $e$ to $E^+$.
            Insert $e$ into $G$;
            Update edge weights in $G$;
            $count += 1$;
        end while
    else
        while $count <= |k|$ do
            Pick the edge $e \in$ EG of the lowest weight;
            Remove $e$ from $G$;
            Add $e$ to $E^-$;
            Update edge weights in $G$;
            $count += 1$;
        end while
    end if
end for
return $E^-, E^+$
```

any value in the range $(0, 1]$. Even when the density value is fixed, which can determine the number of edges to be added/removed as a constant k, it requires enumerating all size-k edge subsets of the edge group of C possible edges, which is of complexity $O\binom{C}{k}$. However, with a fixed density value (and thus a fixed number k of edges to be removed/added), we can show that the objective function (5) is submodular. Then by utilizing the submodular property, we can design a near-optimal approximation solution to choose the k edges for insertion/removal.

Based on this reasoning, FAIRLP is designed as a two-step procedure: (1) *Step 1*: For each edge group (either protected or un-protected), calculate the number of edges to be added/removed to flatten the density across protected and un-protected edge groups; and (2) *Step 2*: for each edge group (either protected or un-protected), pick the specific edges to be added and removed. The pseudo code of FAIRLP is included in Algorithm 1. Next, we explain the details of each step.

Step 1: Calculate number of edges to be added or removed. We consider three alternative solutions to flatten the density of the edge groups: the density of all edge groups can be flattened to be equal to the *average*, the *minimum*, or the *maximum* density of these groups. We name these three methods as **AvgD**, **MinD**, and **MaxD**, respectively. These three methods modify the graph in different ways, and thus affect the prediction accuracy differently. In particular, **MinD** only removes edges from the un-protected edge group (i.e., intra-group edges). It mitigates the accuracy disparity between protected and un-protected groups by reducing the prediction accuracy of the un-protected group. **MaxD** only adds new edges to the protected edge group and mitigates the accuracy disparity by increasing the prediction accuracy of the protected group. Finally, **AvgD** removes edges from the un-protected group and inserts edges into the protected group. It reduces the accuracy disparity by reducing the prediction accuracy of the un-protected group and increasing that of the protected group simultaneously. In the following discussions, we use r to indicate the flattened density, regardless if it is AvgD, MinD, or MaxD.

Let k be the number of edges to be inserted/deleted. To calculate the value of k, first, FAIRLP calculates the density of the protected and un-protected edge groups. Then the flattened density r is computed. Note that r is computed differently by MinD, MaxD, and AvgD. Next, for the protected/un-protected edge group EG, FAIRLP computes its number of possible edges, denoted as $c_{\text{max}}(EG)$. Specifically, given the edge group $EG = \{e(v, v')\}$, its $c_{\text{max}}(EG)$ is calculated as:

$$c_{\text{max}} = \begin{cases} \frac{k_1 \cdot (k_1 - 1)}{2} & \text{if } v.f = v'.f; \\ k_1 \cdot k_2 & \text{Otherwise.} \end{cases} \tag{6}$$

where k_1 and k_2 are the number of nodes that are of value $v.f$ and $v.f'$ respectively. For example, c_{max} value of the M-F edge group is computed as $c_{max} = k_1 \cdot k_2$, where k_1 and k_2 are the number of male and female nodes. Then the number of edges k to be added/removed is calculated as $k = r \cdot c_{\text{max}} - c_{\text{real}}$, i.e., the difference between the expected number of edges ($r \cdot c_{\text{max}}$) and the actual number of edges (c_{real}). If $k > 0$, k new edges are inserted into EG; otherwise, k existing edges are deleted from EG.

Step 2: pick k edges to be added/removed. When k is fixed, we can show that the objective function (5) is submodular. The important implication is that with a fixed density and thus a fixed number k of edges to be removed/added, **continually choosing the edge $e(v_i, v_j)$ with the lowest weight (i.e., the most unimportant edges)**, up to k, near-optimally solves (5) with $(1-\frac{1}{\mathsf{e}})$-approximation ratio, where $\mathsf{e} = 2.718\ldots$ is the base of the natural logarithm. A naive method of picking k most unimportant edges is to sort all edges (including the potential new ones) by their weights, and pick the top-k edges by their weights in ascending order (i.e., the k most unimportant ones) for insertion/removal. However, deleting/inserting edges will change the weights of other edges. For example, consider three nodes v_1, v_2, and v_3 which are connected pairwise. Removing the edge between v_1 and v_2 will change the weight of the edges (v_1, v_3), as v_2 is not the common neighbor of v_1 and v_3 anymore. Therefore, instead of picking the k most un-important edges statically (as the naive method), FAIRLP picks the most unimportant edges in a dynamic way. Specifically, each time after an edge is picked for insertion/deletion, FAIRLPrecalculates the weights for all candidate edges in E^+ and E^-, and sorts these edges again by their updated weights in ascending order. Next, FairLP picks the edge of the lowest weight, and repeats until all k edges are picked. We have the following theorem to show the approximation ratio of Step 2 of FairLP.

Theorem 1. With a fixed value of $density(E^*)$ and $density(\overline{E^*})$, the objective function (5) has $(1 - \frac{1}{\mathbf{e}})$-approximation ratio.

Below we give the proof sketch of Theorem 1. With a given number of edges to be added/deleted, we can show that the objective function in Formula (5) is submodular. Intuitively, sumodularity is a diminishing returns property of the set function. Formally, given a set X, subsets $A \subseteq B \subseteq X$, and any element $x \in X \backslash B$, We define $\Delta_F(x \backslash A)$ as the incremental value gained in the function F by adding x to A:

$$\Delta_F(x \backslash A) = F(A \cup \{x\}) - F(A). \qquad (7)$$

Then F is submodular if $\Delta_F(x \backslash A) \geq \Delta_F(x \backslash B)$ for any A, B. In other words, the incremental value of adding x to the result set diminishes as the result set grows. An important property of the submodular function is that a greedy algorithm that chooses the item with the highest incremental value in iterations yields a solution that is guaranteed to have a $(1 - \frac{1}{\mathbf{e}})$-approximation ratio, where $\mathbf{e} = 2.718\ldots$ is the base of the natural logarithm.

Let $EG_1 \subseteq EG_2 \subseteq EG$ be two subsets of edges in the edge group EG, and let $e \in EG \backslash EG_2$. Consider two cases: (1) if e is included in EG_1; and (2) if e is not included in EG_1. For Case 1, e must also be included in EG_2 given the fact that $EG_1 \subseteq EG_2$. Therefore, when adding e to the edge group of the smallest sum of edge weights, any entry $e' \notin EG_2$ will result in incremental value of at least $\Delta_F(e \backslash EG_1)$ = $w(e)$ (Δ_F is defined in (7)), which is more than the corresponding gain to $\Delta_F(e \backslash EG_2)$, since e is already included in EG_2. In Case 2, where the entities contained within EG_1 and EG_2 are the same, the incremental value $\Delta_F(e \backslash EG_1) = \Delta_F(e \backslash EG_2)$. Therefore, the function F (i.e., the objective function in (5)) is submodular over the edge weights. Therefore, with a fixed density and thus a fixed number k of edges to be removed/added, continually choosing the edge $e(v_i, v_j)$ with the lowest weight, up to k, near-optimally solves (5) with $(1 - \frac{1}{\mathbf{e}})$-approximation ratio.

Evaluation of Disparity Mitigation

In this section, we present the empirical performance evaluation of FairLP in terms of its mitigation effectiveness and impacts on prediction accuracy. We use the same computing environments, datasets, link prediction algorithms, and accuracy evaluation metric as described before. We consider five alternative mitigation approaches for comparison with FairLP. These five approaches are categorized into two types, namely *pre-processing* and *in-processing* mitigation methods. Next, we explain the details of these methods.

Pre-processing baseline methods. We consider two alternative pre-processing mitigation methods: (1) **Preferential sampling (PS)** (Kamiran and Calders 2012): Unlike FairLP that flattens the density of protected and unprotected edge groups, PS modifies the original graph by adding/deleting edges to remove the dependency between the prediction label (0/1) and the edge group membership (i.e., inter- or intra-group edges); (2) **Node-degree weighting based mitigation (ND)**: Similar to FairLP, ND flattens the density of protected and un-protected groups. However, its edge weight is defined based on the degree of both end nodes: where d_i is the degree for node v_i. Intuitively, the edges that connect popular nodes (i.e., nodes of high degree) are more important (and thus have higher weights) than the edges that connect un-popular nodes. We use the function $\log(\cdot)$ to deal with the node degrees at different order of magnitudes. Since $\log(\cdot)$ is invalid for those nodes that are not connected with others (i.e., $d_i = 0$), we add one to all node degrees.

Intuitively, the comparison between FairLP and PS can justify whether the imbalanced density of different groups is indeed the root cause of accuracy disparity, while the comparison between FairLP and ND justifies which node weight scheme (neighborhood based vs. node degree based) better addresses the trade-off between fairness and accuracy.

We also consider three variants of FairLP (i.e., **MinD**, **AvgD** and **MaxD**) that flatten the density of edge groups to the minimum, average, and maximum density of all edge groups respectively. We compare the performance of these three different variants in terms of fairness and accuracy. The results show that **AvgD** best addresses the trade-off between fairness and prediction accuracy. We omit the details of these results due to limited space. In the following discussions we only consider **AvgD** (referred as FairLP for simplicity).

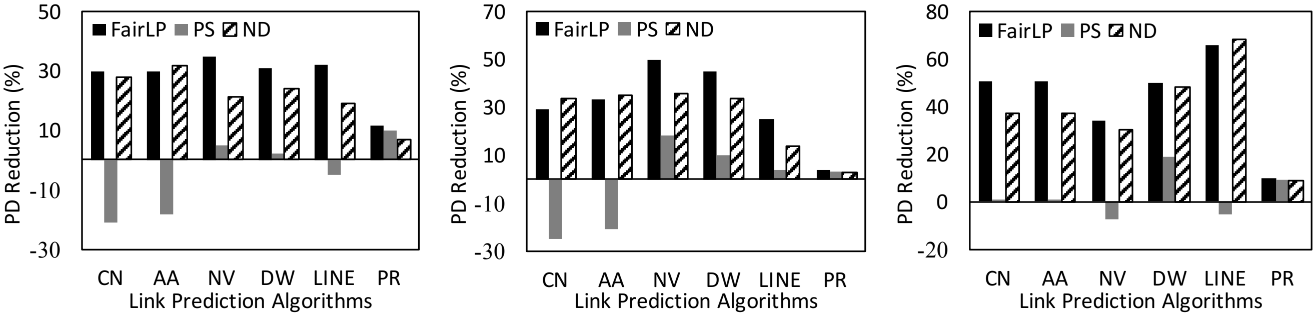
In-processing baseline methods. We consider three fairness-aware link prediction methods as baselines: (1) **FLIP** (Masrour et al. 2020) is an in-processing mitigation algorithm for link prediction. [3]; (2) **Fairwalk** (Rahman et al. 2019) is an in-processing mitigation method that is applied to node2vec embedding (Grover and Leskovec 2016). Then the node2vec embedding is used for link prediction; and (3) **FairPageRank** (Tsioutsiouliklis et al. 2021) is an in-processing mitigation method that is applied to PageRank algorithm. Then the PageRank scores are used for link prediction.

Bias Mitigation Effectiveness

To evaluate the effectiveness of the mitigation methods, we measure the *disparity reduction* R, defined as following: $R = \frac{|\mathrm{D_O}| - |\mathrm{D_F}|}{|\mathrm{D_O}|}$, where $\mathrm{D_O}$ and $\mathrm{D_F}$ indicate the accuracy disparity of the original link prediction (without fairness) and with fairness. We measure and compare both PD and TPD for accuracy disparity. Intuitively, positive R value (i.e., $R > 0$) indicates the effectiveness of mitigation, and the larger the better. Conversely, negative R value (i.e., $R < 0$) indicates the failure of mitigation. We consider the absolute values instead of original values of $\mathrm{D_O}$ and $\mathrm{D_F}$ because they can be negative. Our main observations are listed below.

Disparity reduction by FairLP. We present the results of PD disparity reduction by FairLP for each link prediction algorithm in Figure 2. The results of TPD disparity reduction results are similar; they are thus omitted due

[3] FLIP: https://github.com/farzmas/FLIP



(a) Google+ (b) Facebook (c) DBLP

Figure 2: Disparity reduction (%) (NV: node2vec, DW: DeepWalk, PR: PageRank)

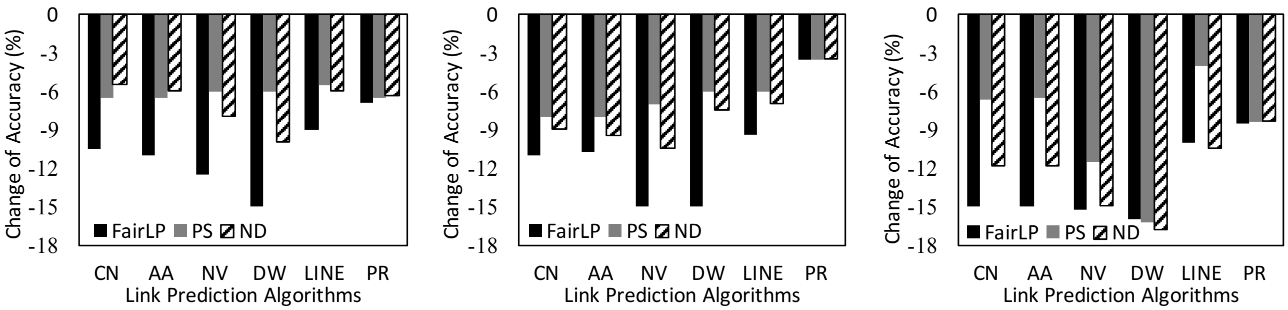
Table 3: Disparity and prediction accuracy: FairLP vs. baselines. The TO columns indicate the trade-off results of TO_T. The best trade-off values per dataset are highlighted in bold.

Category	Mitigation method	Google+				Facebook				DBLP			
		PD	TPD	AUC	TO	PD	TPD	AUC	TO	PD	TPD	AUC	TO
Pre-processing	FairLP + CN	-0.21	-0.21	0.87	**0.24**	-0.15	-0.16	0.86	**0.19**	-0.12	-0.12	0.69	0.17
	FairLP + AA	-0.21	-0.21	0.87	**0.24**	-0.15	-0.17	0.87	**0.19**	-0.12	-0.12	0.69	0.17
	FairLP + node2vec	-0.16	-0.22	0.81	0.27	-0.1	-0.18	0.8	0.23	-0.13	-0.15	0.78	0.19
	FairLP + DeepWalk	-0.16	-0.22	0.79	0.28	-0.11	-0.17	0.8	0.21	-0.086	-0.1	0.75	**0.13**
	FairLP + Line	-0.17	-0.22	0.83	0.27	-0.17	-0.21	0.84	0.25	-0.058	-0.15	0.7	0.21
	FairLP + PageRank	-0.23	-0.23	0.78	0.29	-0.25	-0.24	0.81	0.30	-0.18	-0.18	0.63	0.28
In-processing (Baselines)	FLIP	-0.11	-0.2	0.79	0.25	-0.093	-0.18	0.8	0.23	-0.13	-0.21	0.76	0.28
	Fairwalk	-0.22	-0.24	0.93	0.26	-0.19	-0.23	0.95	0.24	-0.19	-0.2	0.92	0.22
	FairPageRank	-0.25	-0.24	0.67	0.35	-0.24	-0.25	0.67	0.37	-0.19	-0.18	0.67	0.28

to limited space. We observe that FairLP delivers significant disparity reduction for all six link prediction algorithms on the three datasets. In particular, PD disparity reduced by FairLP ranges between [3%, 67%] while TPD disparity reduction ranges between [3%, 51%]. This demonstrates the effectiveness of FairLP in disparity mitigation. We also observe that the PageRank algorithm always receives the smallest disparity mitigation among all link prediction algorithms, as the removed/added edges that are modified by FairLP are few and are least likely to be sampled by the random walk when calculating PageRank scores.

FairLP vs. PS. Recall that unlike FairLP, PS tries to mitigate the data bias by removing the dependency between the prediction labels and the edge group membership. By comparing the results between FairLP and PS in Figure 2, it can be observed that such dependency is not the root cause of disparity, as PS fails to mitigate the disparity in most of the settings. On the other hand, the effectiveness of FairLP proves that the imbalanced group density is one of the causes of accuracy disparity. **Impact of edge weight schemes on disparity reduction.** Both FairLPand ND flatten the group density by removing/adding the same number of edges. Which edges to add/remove depends on the edge weight schemes. Our main observation from Figure 2 is that, on Google+ and Facebook graphs, the performance of ND is similar to FairLP on those neighborhood-based prediction algorithms (i.e., CN (Liben-Nowell and Kleinberg 2007) and AA (Adamic and Adar 2003)), and it loses to FairLPon the graph embedding algorithms (i.e., node2vec (Grover and Leskovec 2016), DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), and Line (Tang et al. 2015)) significantly. On the other hand, the results on DBLP graph are different - ND wins FairLP significantly on CN and AA algorithms, and have similar performance as the graph embedding algorithms. To find out the reason behind such difference in performance across three datasets, we then performed further analysis of edges added/removed by FairLP and ND (denoted as Δ_F and Δ_N). We found that on Google+ and Facebook datasets, Δ_F and Δ_N share a large portion. Regarding the edges in $\Delta_F - \Delta_N$ and $\Delta_N - \Delta_F$, they are of low common neighbor size (thus do not change much of the link prediction based on CN and AA) but different structure (thus different graph embedding). On the other hand, on DBLP dataset, the edges in $\Delta_F - \Delta_N$ and $\Delta_N - \Delta_F$ are of larger common neighbor size (thus impacts much the link prediction based on CN and AA) but more similar structure (thus similar graph embedding).

FairLP vs. baselines. Since both in-processing methods (i.e. FLIP (Masrour et al. 2020) and Fairwalk (Rahman et al. 2019)) do not have accuracy disparity before mitigation, we only report the accuracy disparity of the prediction results. Table 3 (PD and TPD columns) shows the results of accuracy disparity of these mitigation methods. Our first observation is that FairLP outperforms both Fairwalk and FairPageRank in terms of PD and TPD for most of the link prediction algorithms. Note that although Fair-



(a) Google+ (b) Facebook (c) DBLP

Figure 3: Impact on prediction accuracy (AUC) (%): FAIRLP vs. pre-processing methods (NV: node2vec, DW: DeepWalk, PR: PageRank).

Table 4: Impacts of FAIRLP on Network Properties: Network Modularity (Modularity), Eigenvector centrality (EC), Average Node Degree (AND), and Average Clustering Coefficient (ACC)

Metric	Google+				Facebook				DBLP			
	Modularity	EC	AND	ACC	Modularity	EC	AND	ACC	Modularity	EC	AND	ACC
Original network	0.22	0.093	32.49	0.28	0.24	0.058	27.65	0.3	0.094	0.0005	4.49	0.19
FAIRLP	0.005	0.082	30.56	0.1	0.0075	0.05	27.45	0.11	0.034	0.0005	3.91	0.21
PS	0.21	0.093	32.49	0.15	0.24	0.058	27.65	0.19	0.089	0.0005	4.49	0.13
ND	0.024	0.024	30.56	0.11	0.0069	0.023	27.45	0.12	0.0095	0.00009	3.91	0.083

walk is customized for the node2vec algorithm, the performance of Fairwalk is worse than FAIRLP + node2vec. This demonstrates that FAIRLP is effective in disparity mitigation although it is agnostic to the prediction algorithms. Regarding the comparison with FLIP, our observation is that, FAIRLP obtains lower accuracy disparity than FLIP on DBLP dataset. Although FAIRLP does not outperform FLIP on Google+ and Facebook datasets, its performance is still comparable with FLIP.

Impact of FAIRLP on Prediction Accuracy

To evaluate the impact of FAIRLP on prediction accuracy, we evaluate the amounts of *change of accuracy*: $C = \frac{A_F - A_O}{A_O}$, where A_O and A_F indicate the prediction accuracy (AUC or AP) of the original link prediction (without fairness) and with fairness. Intuitively, negative C indicates that link prediction incurs loss on prediction accuracy.

Accuracy loss by FAIRLP. we evaluate the change of accuracy by FAIRLP, and show the change of AUC in Figure 3. The results of change of AP are similar to change of AUC; thus we omit them due to limited space. The results show that FAIRLP incurs accuracy loss to some extent, which is always no more than 16%.

FAIRLP vs. pre-processing methods. Figure 3 shows that FAIRLP brings more accuracy loss than PS and ND except on PageRank algorithm. Such accuracy loss is expected due to the trade-off between accuracy and fairness. The reason why accuracy losses of FAIRLP , PS and ND on PageRank algorithm is similar is that all the three mitigation methods only modify a small number of edges, which unlikely to be sampled by random walk when calculating PageRank scores.

FAIRLP vs. baselines. The results of prediction accuracy of FAIRLP and the three baseline methods are shown in Table 3 (AUC column). First, the prediction accuracy of all the link prediction methods after applying FAIRLP is still acceptable. Second, although Fairwalk always produces the best accuracy, such good accuracy is achieved with the sacrifice of fairness, as it always has the worst performance on accuracy disparity. On the other hand, FLIP has better disparity but worse accuracy than FAIRLP. Furthermore, FAIRLP+PageRank outperforms FairPageRank in terms of prediction accuracy in most of the settings except on DBLP dataset, where both FAIRLP+PageRank and FairPageRank have similar performance in both prediction accuracy and bias mitigation.

Trade-off between Fairness and Accuracy

To measure the trade-off between fairness and prediction accuracy (PD or TPD), we measure the trade-off as follows:

$$TO_P = \frac{|PD|}{AUC},\ TO_T = \frac{|TPD|}{AUC},$$

Intuitively, smaller TO_P/TO_T values indicate better trade-off between fairness and prediction accuracy.

The results of trade-off TO_T of all methods are shown in Table 3 (“TO” columns). The results of TO_P are similar and thus are omitted due to the limited space. We observe that there is always at least one link prediction method whose incorporation with FAIRLP can deliver better trade-off then the baseline methods. Furthemore, for DBLP datasets, the trade-off of each link prediction method in the pre-processing category outperforms all the baseline methods. This demonstrates the superiority of FAIRLP in addressing the trade-off issue between prediction accuracy and its disparity across different groups.

Impacts of FAIRLP on Network Properties

We evaluate four types of network properties: network modularity (Eqn. (3)), *average node degree*, *average clustering coefficient*, and Eigenvector centrality. We measure these four types of network properties before and after applying FAIRLP, PS and ND on the original training network graph. The results are reported in Table 4. First, FAIRLP reduces network homophily significantly. This explains why FAIRLP is effective in reducing accuracy disparity, as network homophily leads to imbalanced group density, which is identified as one of the causes of accuracy disparity. Second, FAIRLP does not change graph characteristics significantly. Furthermore, although PS best preserves the network properties, given its bad performance in accuracy disparity reduction (as it still preserves network modularity), PS fails to remove the true cause of disparity as successfully as FAIRLP.

Discussions

Impact on network homophily by FAIRLP. One important property of FAIRLP is that, for each edge group EG (either protected or non-protected), the total number of edges that are added to EG or removed from EG is inversely proportional to its original density. In other words, the group of lower (higher, resp.) connection ratio will have more (or fewer) edges to be removed. In our setting, more inter-group edges will be added/removed than the intra-group edges. This will reduce the network homophily of the original network graph, and thus leads to the effectiveness of disparity mitigation. The reduction of network homophily is also observed empirically (Table 4).

Types of social networks (un)suitable for FAIRLP. First, as FAIRLP only considers two node groups, it is only suitable to *bi-populated* social networks in which the nodes belong to two different groups. Extending FAIRLP to the *multi-populated* social networks that contain more than two edge groups can start from re-defining the edge groups based on the multiple node groups. Intuitively, given a protected node feature that has ℓ distinct values, there will be $\frac{(\ell+1)\ell}{2}$ edge groups. Regarding the next step of bias mitigation, our intuition is that, balancing the group density across all the edge groups still remains effective for mitigating link prediction accuracy over such multi-populated social network graphs. This can be validated through additional empirical studies.

Second, FAIRLP assumes the network graph contains a well-defined protected attribute (e.g., gender and race) that allows the nodes to be partitioned into groups. However, the protected attribute may not be available in practice due to many reasons (e.g., privacy protection). New definitions of groups and fairness are needed for these cases.

Third, FAIRLP is effective in bias mitigation only when the original network graph has imbalanced group density. Since imbalanced group density is one of the causes of accuracy disparity, FAIRLP may fail to reduce accuracy disparity if all groups have uniform density in the original network already. Thus new studies are needed to identify other causes of accuracy disparity besides imbalanced group density, and new algorithms are in need to remedy these causes.

Effectiveness of FAIRLP under other bias measurements. FAIRLP only measures the disparity in prediction accuracy across different groups. Since FAIRLP reduces network homophily, intuitively, it would be effective to mitigate other types of bias, e.g., disparity in visibility (Fabbri et al. 2020; Stoica and Chaintreau 2018), that are sourced from network homophily. The effectiveness of FAIRLP for these bias measurements can be investigated by further empirical studies.

Conclusion and Future Work

We investigate fairness in link prediction on social network graphs. First, we formalize the fairness definition for link prediction as requiring equal prediction accuracy across different edge groups. Second, we unveil the existence of disparity of link prediction accuracy in a number of state-of-the-art link prediction algorithms. Third, we identify the imbalanced density of different edge groups as one cause of accuracy disparity, and design a pre-processing bias mitigation method named FAIRLP to remove the identified cause. Our experiments demonstrate that FAIRLP better addresses the trade-off between fairness and accuracy compared with the existing fairness-aware link prediction methods.

This paper only considers bias in predicting inter-group links. As homophily is not necessarily equally strong within different groups (Stoica and Chaintreau 2018), in the future, we will also consider bias in within-group link prediction.

Acknowledgements

This work is supported in part by the US National Science Foundation under grants 1948432 and 2047843. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Adamic, L. A.; and Adar, E. 2003. Friends and neighbors on the web. *Social networks* 25(3): 211–230.

Brin, S.; and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30(1): 107–117. ISSN 0169-7552.

Calders, T.; Kamiran, F.; and Pechenizkiy, M. 2009. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*. IEEE.

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

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*.

Fabbri, F.; Bonchi, F.; Boratto, L.; and Castillo, C. 2020. The Effect of Homophily on Disparate Visibility of Minorities in People Recommender Systems. ICWSM.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. KDD.

Freedman, S.; and Jin, G. Z. 2008. Do social networks solve information problems for peer-to-peer lending? Evidence from Prosper. com .

Goh, G.; Cotter, A.; Gupta, M.; and Friedlander, M. P. 2016. Satisfying real-world goals with dataset constraints. NeurIPS, 2415–2423.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. KDD, 855–864.

Guy, I.; and Pizzato, L. 2016. People recommendation tutorial. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 431–432.

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

Hoyer, P.; Janzing, D.; Mooij, J. M.; Peters, J.; and Schölkopf, B. 2008. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems* 21: 689–696.

Kamiran, F.; and Calders, T. 2009. Classifying without discriminating. In *2nd International Conference on Computer, Control and Communication.*

Kamiran, F.; and Calders, T. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33(1): 1–33.

Kang, J.; He, J.; Maciejewski, R.; and Tong, H. 2020. InFoRM: Individual Fairness on Graph Mining. KDD.

Kang, J.; Xie, T.; Wu, X.; Maciejewski, R.; and Tong, H. 2021. MultiFair: Multi-Group Fairness in Machine Learning. *arXiv preprint arXiv:2105.11069* .

Karimi, F.; Génois, M.; Wagner, C.; Singer, P.; and Strohmaier, M. 2018. Homophily influences ranking of minorities in social networks. *Scientific Reports* 8. doi: 10.1038/s41598-018-29405-7.

Lee, E.; Karimi, F.; Wagner, C.; Jo, H.-H.; Strohmaier, M.; and Galesic, M. 2019. Homophily and minority-group size explain perception biases in social networks. *Nature Human Behaviour* 3(10): 1078–1087.

Li, P.; Wang, Y.; Zhao, H.; Hong, P.; and Liu, H. 2020a. On dyadic fairness: Exploring and mitigating bias in graph connections. ICLR.

Li, Y.; Ning, Y.; Liu, R.; Wu, Y.; and Hui Wang, W. 2020b. Fairness of Classification Using Users' Social Relationships in Online Peer-To-Peer Lending. In *Companion Proceedings of the Web Conference 2020*, 733–742. New York, NY, USA.

Liben-Nowell, D.; and Kleinberg, J. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58(7): 1019–1031.

Masrour, F.; Wilson, T.; Yan, H.; Tan, P.-N.; and Esfahanian, A.-H. 2020. Bursting the Filter Bubble: Fairness-Aware Network Link Prediction. AAAI.

McPherson, M.; Smith-Lovin, L.; and Cook, J. M. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27(1): 415–444.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* .

Newman, M. E.; and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical review E* 69(2): 026113.

Nilizadeh, S.; Groggel, A.; Lista, P.; Das, S.; Ahn, Y.-Y.; Kapadia, A.; and Rojas, F. G. 2016. Twitter's Glass Ceiling: The Effect of Perceived Gender on Online Visibility. ICWSM.

Niu, B.; Ren, J.; and Li, X. 2019. Credit scoring using machine learning by combing social network information: Evidence from peer-to-peer lending. *Information* 10(12): 397.

Palowitch, J.; and Perozzi, B. 2019. Monet: Debiasing graph embeddings via the metadata-orthogonal training unit. *arXiv preprint arXiv:1909.11793* .

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

Rahman, T.; Surma, B.; Backes, M.; and Zhang, Y. 2019. Fairwalk: Towards Fair Graph Embedding. In *International Joint Conferences on Artificial Intelligence*, 3289–3295.

Sanz-Cruzado, J.; and Castells, P. 2019. Contact recommendations in social networks. In *Collaborative Recommendations: Algorithms, Practical Challenges and Applications*, 519–569. World Scientific.

Stoica, A.-A.; Riederer, C.; and Chaintreau, A. 2018. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. *WWW* 923–932.

Suresh, H.; and Guttag, J. V. 2019. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002* .

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. WWW.

Tolan, S.; Miron, M.; Gómez, E.; and Castillo, C. 2019. Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in catalonia. ICAIL, 83–92.

Tsioutsiouliklis, S.; Pitoura, E.; Tsaparas, P.; Kleftakis, I.; and Mamoulis, N. 2021. Fairness-Aware PageRank. *The Web Conf* 3815–3826.

Wei, Y.; Yildirim, P.; Van den Bulte, C.; and Dellarocas, C. 2016. Credit scoring with social network data. *Marketing Science* 35(2): 234–258.

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