Fair Collective Classification in Networked Data

Karuna Bhaila
University of Arkansas
Fayetteville, AR 72701, USA
kbhaila@uark.edu

Yongkai Wu
Clemson University
Clemson, SC 29631, USA
yongkaw@clemson.edu

Xintao Wu
University of Arkansas
Fayetteville, AR 72701, USA
xintaowu@uark.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020610

***Abstract*—Collective classification utilizes network structure information via label propagation to improve prediction accuracy for node classification tasks. Because these models use information from previously labeled nodes which often contain historical bias, they may result in predictions that are biased w.r.t. the sensitive attributes of nodes such as race and gender. Throughout inference, this bias may even be amplified due to propagation especially for networks characterized by homophily. Despite past and ongoing research on fair classification, research to ensure fair collective classification still remains unexplored. In this paper, we present a fair collective classification framework (denoted as FairCC) and formulate various heuristic methodologies, including node reweighting, threshold adjustment, and postprocessing, to achieve fair prediction. We also implement and test several naive methodologies for fair collective classification. Experiments on semi-synthetic datasets highlight the insufficiency of the naive methodologies and demonstrate the effectiveness of the proposed heuristics in significantly reducing prediction bias.**

***Index Terms*—collective classification, inference, label propagation, fairness**

I. Introduction

Traditional machine learning models used for classification are often founded on the basis of an underlying assumption that the data are identically and independently distributed (IID). Imposing the same assumption on networked data may lead to loss of useful information regarding entities' influence on each others' attributes/labels. Relational machine learning and collective inference techniques [1]–[3] can be applied in such cases for *within-network classification* which is the process of estimating labels for entities linked to other entities that may or may not be labeled. These joint inference techniques for estimating unknown variables of linked entities simultaneously are referred to as Collective Classification (CC).

Collective classification utilizes the network structure and underlying network properties such as attribute correlation to improve prediction accuracy. This type of correlation is formally defined by the principle of *homophily* which states that contact between similar people occurs at a higher rate than among dissimilar people and can be observed in many real-life networks with varying strength [4]. For example, in a social network, two users with a mutual connection are more likely to have the same political affiliation compared to two random users. Relational learning models either learn or assume the presence of such homophily and propagate known entity labels throughout the network accordingly to make predictions for unlabeled entities. As the unlabeled entities may also have connections among themselves, implementing relational models in a collective manner allows the model to estimate the interrelated values simultaneously.

In recent years, machine learning researchers have been increasingly and rightfully concerned with making automated decision making systems socially and legally fair [5]–[7]. It is important to validate learning methods not just with performance evaluation but also fairness measures to avoid discriminating one social group over others. Although, collective classification approaches may improve prediction accuracy by utilizing network homophily, it is imperative to analyze it through the lens of social inequality to ensure fair predictions for all groups.

From a social science perspective, network structure can amplify discrimination in resource distribution [8], [9]. DiMaggio et al. [8] combined theories from social science and network science to suggest that small initial advantages and disadvantages can develop into greater differences and this phenomenon is amplified when network effects compound the initial endowments at the individual-level through normative influence. They theorize that such amplification occurs in conditions where a network is characterized by homophily, socio-economic characteristics are positively correlated with valuable resources, and an entity is influenced by its network peers to adopt similar practices. Such initial endowment is often observed in most benchmark fair learning datasets in the form of historical bias originating from societal bias. Such bias is also representative of the positive correlation between the privileged social group and the advantageous outcome. Furthermore, collective classification achieves better performance in networks with a higher positive label correlation [10]. The concept of peer influence is realized in CC through label propagation mechanisms which lead to connected individuals being predicted to have the same outcome. Propagation mechanisms in such networks are thus likely to not only propagate but also amplify the unfairness faced by individuals in certain social groups.

To address this issue, we study the problem of fair label propagation and fair inference in collective classification. This paper focuses on the collective classification framework proposed by Macskassy et al. [1]. We extend their univariate collective classification framework to attributed graphs and evaluate unfairness in prediction results for CC models with two different relational classifier components and one collec-

tive inference technique. We empirically verify that collective classification can improve prediction accuracy but also result in unfair estimates w.r.t. the sensitive attribute. To mitigate the resulting bias, we develop a fair collective classification (FairCC) framework. Under the FairCC framework, we formulate various heuristic methodologies by modifying CC components or incorporating bias mitigating techniques. The proposed heuristics build up on existing fair machine learning literature and include an iterative reweighting method based on [11], an iterative threshold adjustment method based on the covariance measure discussed in [12], and postprocessing techniques [11], [13]. Finally we conduct experiments to demonstrate the efficiency of the proposed methodologies.

II. Related Work

A. Fairness in Machine Learning

Several fairness metrics have been proposed to quantify bias in machine learning. They can be broadly categorized into three groups: group fairness which requires equal treatments for groups defined by the protected attribute [6], [7], individual fairness which requires similar treatment or prediction for similar individuals [6], [14], and counterfactual fairness [15], [16] which requires similar prediction for an individual and its counterfactual usually obtained by changing the value of its sensitive attribute. For this work, we focus on the notion of group fairness.

The bias mitigation approaches proposed to achieve group fairness can further be categorized into pre-processing, in-processing, and post-processing techniques based on the stage that they are incorporated into the learning process. Pre-processing techniques are applied directly to training data by modifying labels or attributes or data representations so that the model is trained with unbiased data [11], [14]. In-processing techniques involve a modified objective function that allows algorithm optimization subject to fairness constraints [6], [12], [17]. Post-processing directly changes the predicted labels to ensure fairness [7], [13], [18]. However, most existing algorithms are applied under the assumption that data are IID which may not be directly applicable or effective for non-IID data and methodologies used in collective classification.

B. Fairness in Graph Mining

Recent years have also seen an emergence in the study of fairness in the non-IID setting with graph structured data for various graph mining tasks such as node classification, link prediction, influence maximization, and graph clustering [19], [20]. Most of these works are based on Graph Neural Networks (GNNs) which generate node embedding vectors used in downstream tasks by implementing deep learning approaches on graphs [21]. Fairness in such algorithms is mostly achieved by constraining the objective function with fairness parameters or modifying GNN mechanisms to generate fair node embeddings. However, these GNN based algorithms are computationally expensive and require large volume of data for training & tuning [21]. On the other hand, mechanisms based on label propagation are easier to train with fewer parameters and achieve accuracy comparable to GNNs [22]. Collective classification is even simpler than the C&S method discussed in [22] as it does not require training to optimize any parameters and directly implements label propagation based on certain assumptions on the graph structure. Despite the scalability and expressive power of GNNs, small graphs with favorable structural properties can still benefit from collective classification techniques with minimal overhead.

There have been a few works that explore fairness issues in label propagation mechanisms as well. The authors in [23] investigated fair propagation of node importance to obtain fair node rankings using the PageRank algorithm [24] in contrast to the fair propagation of node labels for fair node classification we study in this work. Our work is most closely related to [25] which studied the effects of network structure and sampling techniques on collective classification performance and bias. The study compared the true positive rates for each label to assess the direction of bias and concluded that bias can be predicted based on class balance and level of homophily in the network. The empirical results showed that minorities may be at a disadvantage when using a small sample from homophilic or neutral networks with class imbalance. However, this work used graphs containing node labels as the only attribute and defined minorities based on class membership rather than membership to a demographic group. Generally accepted notions of group fairness cannot be applied in such cases. Moreover, [25] did not test any fair mechanisms for collective classification. For our work, we analyze unfairness in collective classification on attributed and homophilic networks where each entity is described by a sensitive attribute and some nonsensitive attributes in addition to the labels. We also empirically evaluate various naive as well as heuristic approaches for fair collective classification under two different sampling techniques.

III. Preliminaries

In this section, we describe our formulation and notations for collective classification on an attributed graph. Throughout this paper, without further specifications, calligraphic fonts (e.g., $\mathcal{X}$), bold uppercase letters (e.g., $\mathbf{X}$), bold lowercase letters (e.g., $\mathbf{x}$), and normal lowercase letters (e.g., x) represent sets, matrices, vectors, and scalars, respectively. For any matrix, e.g., $\mathbf{X}$, we use $\mathbf{x}_i$ and $\mathbf{x}_{.,j}$ to denote its i-th row and j-th column respectively. Table I lists the specific notations we use in this paper to formulate collective classification.

A. Collective Classification

We follow the univariate formulation of collective classification [1] where node label is the only attribute and extend it to include sensitive and nonsensitive node attributes. The input is an unweighted, undirected, and attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, Y)$ where $\mathcal{V}$ is a set of N nodes, $\mathcal{E}$ is a set of edges that connect node pairs in $\mathcal{V}$, $\mathbf{X}$ represents the node feature matrix, and Y denotes node labels. Each node v_i is defined by its feature vector $\mathbf{x}_i \in \mathbb{R}^d$ which

TABLE I: Notation

Notation	Definition
$\mathcal{G}$	the entire graph
N	the number of nodes in $\mathcal{G}$
$\mathcal{V}, \mathcal{E}$	a set of nodes/edges
$\mathcal{V}^K, \mathcal{V}^U$	a set of nodes with known/unknown labels
$\mathbf{X}, \mathbf{x}_i$	features of all nodes/the i-th node
S, s_i	the sensitive feature(s) of all nodes/the i-th node
Y, y_i	the label(s) of all nodes/the i-th node
$\hat{Y}, \hat{y}_i$	the predicted label(s) of all nodes/i-th node in $\mathcal{V}^U$
$\mathcal{N}_i$	the 1-hop neighborhood of node the i-th node
$P(y_i = c\|\mathcal{N}_i)$	the cond. prob. of the i-th node belonging to class c given its neighborhood
$\hat{\mathbf{C}}, \hat{\mathbf{c}}_i$	label probabilities of all nodes/i-th node in $\mathcal{V}^U$
$\hat{c}_{i,j}$	the probability of the i-th node belonging to class j
Δ_{SP}	statistical parity based unfairness measure
$f_L(\cdot)$	the local classifier
$f_R(\cdot)$	the relational classifier
$f_{wR}(\cdot)$	the relational classifier that incorporates node reweighting

includes a sensitive attribute s_i. We use $\mathbf{x}_i \setminus s_i$ to denote its nonsensitive features. Node label y_i is known only for a subset of nodes in $\mathcal{V}$. We denote this subset of nodes as $\mathcal{V}^K$ and their corresponding attributes as $\mathbf{X}^K$, sensitive attributes as S^K, and labels as Y^K. The remaining nodes are denoted as $\mathcal{V}^U$ and corresponding notations follow accordingly. The task then is to simultaneously infer the values y_i for $v_i \in \mathcal{V}^U$ or a probability distribution over those label values. We use $\hat{Y}$ to denote the predicted labels and $\hat{\mathbf{C}}$ to denote a matrix containing vectors of label probabilities for each node in $\mathcal{V}^U$. $\hat{\mathbf{c}}_i$ refers to the i-th row vector $\hat{\mathbf{C}}$ containing the label probabilities of an arbitrary node v_i and $\hat{c}_{i,j}$ refers to the element in i-th row and j-th column in $\hat{\mathbf{C}}$ containing the probability of node v_i being classified into the j-th class. For simplicity, we assume both s and y to be binary and consider only undirected and unweighted edges between arbitrary node pairs. In this setting, $\hat{\mathbf{c}}_{\cdot,1}$ denotes the predicted positive class probability (assuming $y = 1$ to be the advantaged outcome). We further use c and a to denote an arbitrary node label and sensitive attribute value respectively.

The collective classification framework comprises three main components: the local classifier, the relational classifier, and the collective inference which are discussed below.

1) Local Classifier: The local classifier f_L estimates label probabilities $P(y_i|\mathbf{x}_i)$ using only node attributes $\mathbf{x}_i$. Initial labels can then be obtained using the optimal decision rule, $y_i = \underset{c}{\operatorname{argmax}} P(y_i = c|\mathbf{x}_i)$. The classifier is trained with $\mathcal{V}^K$ and the goal is to calculate probability estimates or labels and initialize nodes in $\mathcal{V}^U$ with the computed estimates or labels. These initializations are used by the relational classifier to estimate probabilities for nodes whose neighborhood contains some nodes from $\mathcal{V}^U$. For graphs without relevant node attributes, we can initialize $\mathcal{V}^U$ nodes using class prior computed from $\mathcal{V}^K$ or *null* values. In the latter case, the relational classifier simply skips the *null* labeled neighbor nodes when computing a target estimate.

2) Relational Classifier: The relational classifier f_R leverages graph structure and graph properties to directly estimate unknown node labels or a probability distribution over them. These classifiers either learn or assume homophily in the given graph and propagate labels throughout the graph. Instead of estimating the full joint probability $P(Y^U|\mathcal{G})$, the learning process is made simpler with a first-order Markov assumption: $P(y_i|\mathcal{G}) = P(y_i|\mathcal{N}_i)$ where $\mathcal{N}_i$ defines a set of 1-hop neighbors of node v_i such that $P(y_i|\mathcal{N}_i)$ is independent of $\mathcal{V} \setminus \mathcal{N}_i$ [1]. Then a relational model based on $\mathcal{N}_i$ can be used to estimate y_i. We describe two such relational models used in this paper below.

Weighted-Vote Relational Neighbor (WVRN) [1] is the simplest relational classifier; it does not learn any network properties but assumes the existence of homophily in order to estimate node probabilities by setting a node's prediction to be the majority label of its neighbors. For each $v_i \in \mathcal{V}^U$, WVRN estimates $P(y_i|\mathcal{N}_i)$ as the mean of the label probabilities of the entities in $\mathcal{N}_i$.

$$P(y_i = c|\mathcal{N}_i) = \frac{1}{Z} \sum_{v_j \in \mathcal{N}_i} P(y_j = c|\mathcal{N}_j), \tag{1}$$

where Z is the usual normalizer and we omit the edge weight term as we only consider unweighted or uniformly weighted edges.

Relational Bayes (NBR) [26] uses multinomial naive Bayesian classification based on the classes of v_i's neighbors and attributes of v_i with an independence assumption between its neighbors and attributes. A univariate formulation of this relational classifier only uses v_i's neighbors' labels/estimates [1]. We use the univariate formulation to compute a target node's estimate as:

$$P(y_i = c|\mathcal{N}_i) = \frac{P(\mathcal{N}_i|y_i = c)P(y_i = c)}{P(\mathcal{N}_i)}. \tag{2}$$

$P(\mathcal{N}_i)$ denotes the probability of observing an arbitrary node v_i's neighborhood and is the same regardless of v_i's label estimate since the graph is a fixed structure; normalization across the classes allows us to avoid explicitly computing this term. $P(y_i = c)$ is simply the class prior, i.e., the fraction of nodes in $\mathcal{V}^K$ that are labeled c. Assuming independence between all neighbor classes of a node v_i, the neighborhood class distribution, $P(\mathcal{N}_i|y_i = c)$, is given as:

$$P(\mathcal{N}_i|y_i = c) = \frac{1}{Z} \prod_{v_j \in \mathcal{N}_i} P(y_j = \gamma|y_i = c) \tag{3}$$

where Z is a normalizing constant and γ is an arbitrary neighbor label. We compute $P(y_j = \gamma|y_i = c)$ from $\mathcal{V}^K$ as:

$$P(y_j = \gamma|y_i = c) = \frac{\sum\limits_{v_i \in \mathcal{V}^K} \left[\mathbb{I}(y_i = c) \sum\limits_{v_j \in \mathcal{N}_i \setminus \mathcal{V}^U} \mathbb{I}(y_j = \gamma) \right]}{\sum\limits_{\gamma'} \sum\limits_{v_i \in \mathcal{V}^K} \left[\mathbb{I}(y_i = c) \sum\limits_{v_j \in \mathcal{N}_i \setminus \mathcal{V}^U} \mathbb{I}(y_j = \gamma') \right]} \tag{4}$$

where $\mathbb{I}$ refers to an indicator function. Essentially, the neighborhood class distribution term encodes the level of

Algorithm 1 Collective classification (CC)

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, Y), f_L, f_R$
Output: $\{\hat{y}_i\}$

$\hat{\mathbf{c}}_i^{(0)} \leftarrow f_L(\mathbf{x}_i)\ \ \forall v_i \in \mathcal{V}^U$
while $t < T$ **do**
  $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \beta^{(t+1)} \cdot f_R(v_i)^{(t)} + (1 - \beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)}\ \ \forall v_i \in \mathcal{V}^U$
  use $\hat{\mathbf{c}}_i^{(t+1)}$ to obtain $\hat{y}_i^{(t+1)}$
  if $\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)}\ \ \forall i \in \mathcal{V}^U$ **then**
    break
  end if
end while
return $\{\hat{y}_i^{(t+1)}\}$

homophily observed in the subgraph induced by $\mathcal{V}^K$ nodes. Following [1], we consider undirected links and apply Laplace smoothing to account for possible zeros in the estimation of neighborhood class distribution.

3) Collective Inference: A relational model estimates v_i's label based on its neighborhood $\mathcal{N}_i$. However, $\mathcal{N}_i$ may contain nodes from $\mathcal{V}^K$ as well as $\mathcal{V}^U$, i.e., $\mathcal{N}_i = \mathcal{N}_i^K \cup \mathcal{N}_i^U$. The labels for nodes in $\mathcal{N}_i^U$ are also estimated using the relational model. It then follows that just as $\mathcal{N}_i$ influences the estimate of y_i, y_i also influences the estimate of the labels of nodes in $\mathcal{N}_i^U$ since v_i is included in the neighborhood of each node $v_j \in \mathcal{N}_i^U$. Collective inference methods can be implemented in this case to simultaneously estimate these interdependent values. Macskassy et al. explored three different collective inference techniques namely relaxation labeling, Gibbs sampling, and iterative classification [1]. For our work, we focus on relaxation labeling which has been shown to have the best performance in terms of accuracy and time [10].

Relaxation Labeling [1] is an iterative update method that updates the probability estimates for all $v_i \in \mathcal{V}_U$ at step $t+1$ based on estimations obtained from the relational classifier at step t. Relaxation labeling implemented in this manner was sometimes observed to oscillate between two or more graph states. Macskassy et al. [1] imposed simulated annealing during the update step in relaxation labeling by giving more weight to a node's own current estimate and less to the influence of its neighbors during each subsequent iteration to counter this oscillation. The update step is then formulated as:

$$\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \beta^{(t+1)} \cdot f_R(v_i)^{(t)} + (1 - \beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)} \qquad (5)$$

where, $\hat{\mathbf{c}}_i$ is a vector of probabilities representing f_R's estimates of $P(y_i|\mathcal{N}_i)$ and $\beta^0 = k$, $\beta^{(t+1)} = \alpha\beta^{(t)}$, k is a constant between 0 and 1, and α is a decay constant.

Algorithm 1 outlines the collective classification framework. Given a graph $\mathcal{G}$ with labels for nodes in $\mathcal{V}^K$, a local classifier f_L, and a relational classifier f_R, collective classification simultaneously infers the probability estimates for nodes in $\mathcal{V}^U$. First, the nodes in $\mathcal{V}^U$ are initialized with estimates/labels obtained from f_L. Then, the predicted probability for each $v_i \in \mathcal{V}^U$ is computed using the relational model f_R. These probabilities are used to update node label predictions during each iteration of inference.

B. Fairness Metrics

For evaluating the collective classification model, we use statistical parity [6], a widely accepted notion of group fairness measured in terms of the binary sensitive attribute $s \in \{0,1\}$, binary observed label $y \in \{0,1\}$, and the binary predicted label $\hat{y} \in \{0,1\}$. Statistical parity requires that the predictions $\hat{y}$ be independent of the sensitive attribute s, i.e., $\hat{y} \perp s$. It can be formally written as $P(\hat{y}|s = 0) = P(\hat{y}|s = 1)$. We apply the metric, $\Delta_{SP} = |P(\hat{y}|s = 0) - P(\hat{y}|s = 1)|$, to quantitatively evaluate statistical parity.

IV. Fair Collective Classification

In this section we present our fair collective classification (FairCC) and discuss different approaches to achieve fairness. Fairness in collective classification can be evaluated at three different stages throughout the process: local fairness for the predictions obtained from the local classifier, relational fairness for predictions obtained from the relational classifier, and aggregated fairness for final predictions obtained after the last iteration of inference. We will use statistical parity measure for all stages and denote them as $\Delta_{f_L}, \Delta_{f_R}$, and Δ_C. Our goal ultimately is to ensure fairness guarantee in Δ_C.

One naive way of formulating FairCC is to ensure that the local classifier f_L computes fair initial label estimates for nodes in $\mathcal{V}^U$. Since these fair initializations are propagated throughout the graph, it is reasonable to expect that imposing Δ_{f_L}-fairness in the local classifier could help lead to overall fairness. However, a fair f_L that estimates fair labels/probabilities for nodes in $\mathcal{V}^U$ cannot guarantee fairness for each node in $\mathcal{V}^U$ because of potential distribution difference between $\mathcal{V}^K$ and $\mathcal{V}^U$. Moreover, the predictions from fair local classifier are used by the collective classification model only during the first iteration of inference and cannot ensure Δ_{f_R}-fairness in subsequent iterations. We empirically validate this claim in Section V. As bias amplification mainly occurs in the relational classifiers where known and predicted labels are propagated throughout the graph; ensuring Δ_{f_R}-fairness at every iteration of inference can lead to Δ_C-fairness. We now propose methodologies that make f_R predictions fair so that estimates obtained in every iteration are relatively fair w.r.t. the sensitive attribute.

A. Fair Collective Classification via Node Reweighting

The node reweighting (NR) method is based on the reweighting method [11] which aims to mitigate historical bias in data through preprocessing. This method addresses data imbalance in training data by assigning a higher weight to positively classified examples in the unprivileged group compared to positively classified examples in the privileged group while maintaining overall positive class probability. Similarly, it assigns a lower weight to negatively classified examples in the unprivileged group compared to negatively classified examples in the privileged group. The learning model is then

Algorithm 2 Fair Collective Classification via Node Reweighting (FairCC-NR)

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, Y), f_L, f_{wR}$
Output: fair $\{\hat{y}_i\}$

$\hat{\mathbf{c}}_i^{(0)} \leftarrow f_L(\mathbf{x}_i)\ \ \forall v_i \in \mathcal{V}^U$
$w_{c,a}^{(0)} \leftarrow 1\ \ \forall c, \forall a$
while $t < T$ **do**
  $w_{c,a}^{(t+1)} = \beta^{(t+1)} \cdot \frac{P(y=c)^{(t)}}{P(y=c|s=a)^{(t)}} + (1-\beta^{(t+1)}) \cdot w_{c,a}^{(t)}$
  $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \beta^{(t+1)} \cdot f_{wR}(v_i)^{(t)} + (1-\beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)}\ \ \forall v_i \in \mathcal{V}^U$
  $\{f_{wR}$ refers to (IV-A) or (IV-A)$\}$
  use $\hat{\mathbf{c}}_i^{(t+1)}$ to obtain $\hat{y}_i^{(t+1)}$
  compute $\Delta_{SP}^{(t+1)}$
  if $\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)}\ \ \forall i \in \mathcal{V}^U$ and $\Delta_{SP}^{(t+1)} \leq \delta$ **then**
    break
  end if
end while
return $\{\hat{y}_i^{(t+1)}\}$

trained with the reweighted training data allowing it to learn parameters from fair and balanced training data to possibly achieve fairness in the test data.

We propose to incorporate this reweighting technique in the collective classification model during each iterative update to assign weights to nodes for each possible combination of s and y. As opposed to reweigh the training set [11], we derive these sensitive attribute-based weights for all nodes in the graph and update them iteratively. We compute the weights using true labels for nodes in $\mathcal{V}^K$ and predicted labels in $\mathcal{V}^U$. This allows the computed weights to capture the bias in the known labels for $\mathcal{V}^K$ as well as the predicted labels from the previous iteration for $\mathcal{V}^U$ since both types of labels are used by the relational classifiers. The calculated weights are then propagated throughout the graph along with the inferred probabilities/labels. Algorithm 2 shows the pseudo-code of our fair collective classification via node reweighting. In our setup with binary s and binary y, we derive these weights for four combinations of s and y. We start by computing weights based on the known labels and predicted labels obtained from f_L.

$$w_{c,a}^{(0)} := \frac{P(y=c)}{P(y=c|s=a)} \tag{6}$$

Then for each iteration of inference, we derive weights using known labels and predictions obtained from f_R in the previous iteration as shown in Algorithm 2 until the predictions converge to a fair and stable state. Further, since the inference procedure incorporates simulated annealing in the update step to ensure label convergence, we perform a similar update for the node weights by allowing the node weights from the previous iteration to have the same amount of influence as the probability estimates from the previous iteration.

$$w_{c,a}^{(t+1)} = \beta^{(t+1)} \cdot \frac{P(y=c)^{(t)}}{P(y=c|s=a)^{(t)}} + (1-\beta^{(t+1)}) \cdot w_{c,a}^{(t)} \tag{7}$$

We leverage the homophilic properties of networks and apply these weights in the relational classifiers such that positively labeled neighbor nodes from the privileged group have a smaller influence than the negatively labeled neighbor nodes from the same group. The opposite effect is true for neighbor nodes from the unprivileged group. We further control this influence by using the target node's own sensitive attribute-based weight as well. For networks where intra-group edges are denser than inter-group edges, the weights have the effect of increasing the positive predicted probability of nodes in the unprivileged group while decreasing that of the privileged group. Furthermore we compute and update these weights in every iteration using predicted labels from the previous iteration so that the model can adjust for any bias amplification that occurred during label propagation in the previous iteration. We now discuss how we incorporate these node weights into each of the relational classifiers used in this paper.

WVRN incorporates the computed weights by applying them directly to the neighbor's probability based on the neighbor's sensitive attribute and known label or label predicted in the previous iteration. We also apply the weight based on the node's own sensitive attribute and predicted node label.

$$P(y_i = c|\mathcal{N}_i) = \frac{w_{c,s_i}}{Z} \sum_{v_j \in \mathcal{N}_i} P(y_j = c|\mathcal{N}_j) w_{c,s_j} \tag{8}$$

where Z is the usual normalizer, w_{c,s_j} is the weight due to neighbor v_j's sensitive attribute and label to be estimated, and w_{c,s_i} is the weight due to node v_i's own sensitive attribute and the label to be estimated. We normalize the weights before using them to compute the probability estimates.

We similarly modify the NBR classifier by using the node weights according to its own and its neighbors' known/predicted labels and sensitive attributes.

$$P(y_i = c|\mathcal{N}_i) = P(y_i = c)\frac{1}{Z}\Big(\prod_{v_j \in \mathcal{N}_i} (P(y_j = \gamma|y_i = c))^{w_{\gamma,s_j}}\Big)^{w_{c,s_i}} \tag{9}$$

where w_{γ,s_j} is the weight due to neighbor v_j's known/predicted label and sensitive attribute. We scale the values between positive constants a and b such that $1 \leq a < b$ before applying them.

B. Fair Collective Classification via Threshold Adjustment

We now propose a second heuristic, threshold adjustment (TA), to make f_R predictions fair. TA derives fair classification thresholds for each group defined by the sensitive attribute. In our case, we derive two fair thresholds: τ_+ for the privileged group and τ_- for the unprivileged group. We compute these fair thresholds in every iteration so that the predictions obtained by applying these thresholds are relatively fair for that iteration. The threshold adjustment method is based on the covariance-based measure of unfairness [12], which quantifies the dependence between sensitive attributes and classifier predictions as a measure of covariance between the sensitive attribute and the signed distance of the node's

Algorithm 3 Fair Collective Classification via Threshold Adaptation (FairCC-TA)

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, Y), f_L, f_R$
Output: fair $\{\hat{y}_i\}$

$\hat{\mathbf{c}}_i^{(0)} \leftarrow f_L(\mathbf{x}_i)\ \ \forall v_i \in \mathcal{V}^U$
Initialize $\tau_+ = \tau_- = 0.5$
while $t < T$ **do**
  $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \beta^{(t+1)} \cdot f_R(v_i)^{(t)} + (1-\beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)}\ \ \forall v_i \in \mathcal{V}^U$
  compute $\epsilon^{(t+1)}$ {Refer to Section IV-B}
  $\tau_+ \leftarrow \tau_+ + \beta^{(t+1)} \cdot \epsilon^{(t+1)} + (1-\beta^{(t+1)}) \cdot \epsilon^{(t)}$
  $\tau_- \leftarrow \tau_- - \beta^{(t+1)} \cdot \epsilon^{(t+1)} - (1-\beta^{(t+1)}) \cdot \epsilon^{(t)}$
  use $\hat{\mathbf{c}}_i^{(t+1)}$ and τ_+ to obtain $\hat{y}_i^{(t+1)}\ \ \forall i \in \mathcal{V}^U : s_i = +$
  use $\hat{\mathbf{c}}_i^{(t+1)}$ and τ_- to obtain $\hat{y}_i^{(t+1)}\ \ \forall i \in \mathcal{V}^U : s_i = -$
  compute $\Delta_{SP}^{(t+1)}$
  if $\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)}\ \ \forall i \in \mathcal{V}^U$ and $\Delta_{SP}^{(t+1)} \leq \delta$ **then**
    break
  end if
end while
return $\{\hat{y}_i^{(t+1)}\}$

feature vector from the decision boundary for linear classifiers. Minimizing this covariance measure then becomes analogous to minimizing the statistical parity measure.

The collective classification model predicts label probabilities by aggregating neighborhood labels or probabilities; the decision boundary can be interpreted to be the optimal decision function applied over the predicted probabilities. The decision function in this case assigns node labels based on the threshold value τ which is generally set to be 0.5. Unfairness can then be computed as the covariance between the sensitive attributes $\mathbf{s}$ and the difference in the predicted positive class probability $\hat{\mathbf{c}}_{\cdot,1}$ (assuming $y = 1$ to be the advantaged outcome) and the threshold value τ,

$$\text{Cov} = \frac{1}{n(\mathcal{V}^U)} \sum_{v_i \in \mathcal{V}^U} (s_i - \mu_{\mathbf{s}}) \cdot (\hat{c}_{i,1} - \tau) \tag{10}$$

where $n(\mathcal{V}^U)$ refers to the number of nodes in $\mathcal{V}^U$ and $\mu_{\mathbf{s}}$ is the mean of sensitive attribute values for nodes in $\mathcal{V}^U$. Since we derive two different thresholds for each group defined by the sensitive attribute, we decompose (IV-B) as:

$$\text{Cov} = \frac{1}{n(\mathcal{V}^U)} \Bigg[\sum_{v_i \in \mathcal{V}^U : s_i = +} (s_i - \mu_{\mathbf{s}}) \cdot (\hat{c}_{i,1} - \tau_+) + \sum_{v_i \in \mathcal{V}^U : s_i = -} (s_i - \mu_{\mathbf{s}}) \cdot (\hat{c}_{i,1} - \tau_-) \Bigg] \tag{11}$$

Intuitively, we can reason that the optimal fair threshold for the privileged group should be greater than the unfair threshold and the optimal fair threshold for the unprivileged group should be lesser than the unfair threshold, i.e., $\tau_+^* > \tau_+$ and $\tau_-^* < \tau_-$. Let ϵ be the offset value needed to derive the fair thresholds; $\tau_+^* = \tau_+ + \epsilon$ and $\tau_-^* = \tau_- - \epsilon$. We can derive the value of this offset term using (IV-B) such that the computed covariance becomes 0.

$$\epsilon = \frac{-\text{Cov} \cdot n(\mathcal{V}^U)}{\sum_{v_i \in \mathcal{V}^U : s_i = +} (s_i - \mu_{\mathbf{s}}) - \sum_{v_i \in \mathcal{V}^U : s_i = -} (s_i - \mu_{\mathbf{s}})} \tag{12}$$

We compute ϵ and update these group specific thresholds in each iteration after the probability update step as shown in Algorithm 3. This allows us to iteratively approximate the optimal fair thresholds. We do not make any modifications to the relational classifiers and only adjust the group specific thresholds based on the obtained probabilities.

C. Fair Collective Classification via Post Processing

Another approach to reduce bias in collective classification is to directly reduce Δ_C-unfairness by means of modifying final collective classification predictions via postprocessing. In this paper, we pair the collective classification model with two different postprocessing techniques as described below.

a) Reject Option Classification (ROC): is a classifier-agnostic postprocessing method that exploits the low-confidence region of a probabilistic classifier for discrimination reduction [13]. We adopt the same method and apply it over the predictions from the last iteration of collective inference. The ROC method first defines a critical region composed of nodes for which $\max[\hat{c}_{i,1}, 1-\hat{c}_{i,1}] \leq \theta$ (assuming $y = 1$ to be the advantaged outcome). The nodes in the critical region are considered to be influenced by bias and are labeled based on their sensitive attribute values: advantaged outcome for the unprivileged group and disadvantaged outcome for the privileged group. The nodes outside the critical region are classified according to the standard decision rule. θ is a hyper-parameter and chosen according to label probabilities of nodes in $\mathcal{V}^K$ to maximize model performance and minimize bias.

b) Label Flipping (LF): is another classifier-agnostic postprocessing method which directly changes the labels based on estimates from a probabilistic classifier [11]. We first partition nodes from $\mathcal{V}^U$ into two sets, one containing positively classified nodes from the privileged group and the other containing negatively classified nodes from the unprivileged group. We rank the nodes based on their positive class probability in increasing order for the former and decreasing order for the latter. We then flip the predicted labels for an equal number of nodes in both sets to achieve fairness.

Both of these postprocessing techniques can be applied over final CC predictions without modifying the local classifier, the relational classifier, and the inference method.

V. Experiments

A. Data

For the purpose of this study, we derive semi-synthetic datasets from two benchmarks for fair machine learning. 1) The German credit dataset [27] contains information about clients at a German bank and the prediction task is to classify clients as good or bad customer. We use `gender` as the sensitive attribute. 2) The Student dataset [28] describes student

TABLE II: Dataset Statistics

Dataset	German	Student
# of nodes	955	577
# of edges	19,980	8,411
# of node attributes	32	38
density	0.04	0.05
assortativity in y	0.667	0.649
assortativity in s	0.595	0.573

achievements in Portuguese subject at two Portuguese schools. We use the attribute `G3` as the label by categorizing it into < 10 and ≥ 10 groups and treat `sex` as the sensitive attribute.

For both datasets, we manually generate edges based on instance similarity [27] by calculating the weighted Euclidean distance between any arbitrary pair of nodes (v_i, v_j) as $\left(1 + \sqrt{\sum_k w_k (x_{i,k} - x_{j,k})^2}\right)^{-1}$. We include `class` in the Euclidean distance and assign a higher weight to it in order to maintain a homophilous network which is the focus of this paper. We then select the top n node pairs to form undirected and unweighted edges for the graph and also remove any isolated nodes. The value of n depends on the desired value of network density and network homophily.

Table II shows the statistics for both datasets where assortativity in y indicates the degree of label-based homophily, and assortativity in s indicates the degree of sensitive attribute-based homophily in the network. For each dataset, we choose nodes in $\mathcal{V}^K$ using two sampling techniques: random sampling and degree-sorted sampling. For random sampling, we randomly choose 30% of nodes to form $\mathcal{V}^K$. For degree-sorted sampling, we choose the top 30% of nodes as $\mathcal{V}^K$ from a list of all nodes sorted in the descending order of their degrees. The degree-sorted sampling technique samples $\mathcal{V}^K$ such that it contains the central nodes from most clusters in cases where the dataset is comprised of clusters instead of a single connected graph. The two sampling techniques allow us to compare model performance and fairness under different local neighborhood structures for nodes in $\mathcal{V}^K$.

B. Compared Methods

We evaluate and analyze the fair collective classification formulations discussed in Section IV including FairCC-NR-W, FairCC-NR-N, FairCC-TA-W, FairCC-TA-N, FairCC-ROC-W, FairCC-ROC-N, FairCC-LF-W, and FairCC-LF-N. Note that the suffixes -W and -N denote WVNR and NBR relational models, respectively. We compare these methodologies with vanilla collective classification methods, CC-W and CC-N, and naive fair collective classification methods, FairNB+CC-W and FairNB+CC-N. We also compare against the non-collective classification method, NB, and its fair version denoted as FairNB. We do not compare against any GNN mechanisms which utilize graph structure for parameter optimization since the relational classification based methods rely only on propagation without optimization. All methods evaluated in this section are implemented on the basis of AIF360 [29].

- **NB**. An attribute-only Naive Bayes classifier that uses only node features without edge information.
- **FairNB**. A fair Naive Bayes model that incorporates the exponentiated reduction algorithm discussed in [17]. The reduction method reduces fair classification to a sequence of cost-sensitive classification problems and returns a randomized classifier with the lowest empirical error subject to fair classification constraints. Similar to NB, this model does not utilize edge information.
- **CC-W/CC-N.** Collective classification with WVRN or NBR as the relational classifier and relaxation labeling as the inference method originally proposed in [1] performs univariate collective classification and utilizes only the graph structure and node labels. We extend this formulation by allowing the local classifier to make predictions based on node attributes so that the collective classification model as a whole utilizes both node features and edges. Following [26], we use Naive Bayes (NB) as the local classifier.
- **FairNB+CC-W/FairNB+CC-N**. A naive fair CC formulation discussed in Section IV where the local classifier NB is substituted with a fair local classifier FairNB. The relational classifier is WVRN or NBR.
- **FairCC-NR-W/FairCC-NR-N**. FairCC via Node Reweighting (NR) as proposed in Section IV-A for WVRN or NBR relational classifiers.
- **FairCC-TA-W/FairCC-TA-N**. FairCC via Threshold Adjustment (TA) as proposed in Section IV-B for WVRN or NBR relational classifiers.
- **FairCC-ROC-W/FairCC-ROC-N**. FairCC paired with ROC as discussed in Section IV-C.
- **FairCC-LF-W/FairCC-LF-N**. FairCC paired with LF as discussed in Section IV-C.

For FairCC with NR or TA, the fairness threshold δ serves more as an additional check for early stop than a hyperparameter and we use the generally accepted threshold of 0.05 for δ in our experiments. Following the experimental setup in [1], we set $k = 1$, $\alpha = 0.99$ and run the inference procedure for a maximum of 100 iterations. An analysis of the results showed that FairCC methods required comparatively more iterations for convergence of predictions compared to CC but the predictions for both methods generally converged in less than 100 iterations. For random node sampling, we run the experiments 5 times with different random seeds and report their average and standard deviation. For degree-sorted sampling, we run the experiments once and report the evaluation values. We use accuracy (higher is better) and F1-score (higher is better) to evaluate performance and statistical parity (lower is better) to evaluate fairness of the models.

C. Results

Tables III and IV summarize our experiment results on the German and Student datasets respectively with the best performance highlighted in bold font. For each dataset, we consider both degree-sorted sampling and random node sampling. From the tables, we draw the following conclusions.

TABLE III: Performance comparison for non-CC, CC, and FairCC methodologies for German dataset under random and degree-sorted sampling; fairness threshold δ = 0.05 for all bias mitigating methods

Method	Degree			Random		
	ACC(%)↑	F1(%)↑	$\Delta_{\mathbf{SP}}\downarrow$	ACC(%)↑	F1(%)↑	$\Delta_{\mathbf{SP}}\downarrow$
NB	62.93	59.71	0.17	67.10±2.37	62.87±1.20	0.19±0.07
FairNB	62.18	57.21	0.07	67.37±2.74	62.86±1.25	0.03±0.01
CC-W	**84.60**	**80.79**	0.07	86.89±0.79	81.48±1.01	0.10±0.02
FairNB+CC-W	**84.60**	**80.79**	0.07	86.89±0.79	81.48±1.01	0.10±0.02
FairCC-ROC-W	84.45	80.70	0.09	**89.55**±1.05	**85.78**±1.32	0.06±0.06
FairCC-LF-W	**84.60**	**80.79**	0.05	87.19±0.65	81.91±0.81	0.05±0.00
FairCC-NR-W	81.76	76.57	**0.03**	87.57±1.28	82.56±2.16	**0.02**±0.01
FairCC-TA-W	84.45	80.57	0.04	87.84±1.16	83.03±1.51	0.06±0.02
CC-N	75.49	67.28	0.12	91.05±2.31	89.08±2.80	0.19±0.06
FairNB+CC-N	76.53	68.86	0.12	93.74±2.83	92.34±3.43	0.13±0.05
FairCC-ROC-N	76.08	68.33	0.12	92.19±2.30	90.64±2.74	0.10±0.03
FairCC-LF-N	77.88	70.47	0.04	91.29±1.54	89.38±1.78	0.05±0.00
FairCC-NR-N	80.57	74.06	**0.01**	**93.95**±2.81	**92.57**±3.37	**0.04**±0.02
FairCC-TA-N	**80.87**	**74.57**	**0.01**	63.89±5.16	55.90±6.15	0.08±0.04

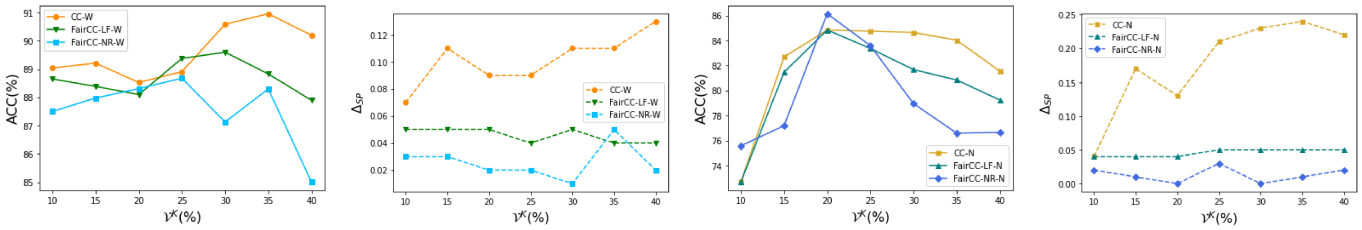
TABLE IV: Performance comparison for non-CC, CC, and FairCC methodologies for Student dataset under random and degree-sorted sampling; fairness threshold δ = 0.05 for all bias mitigating methods

Method	Degree			Random		
	ACC(%)↑	F1(%)↑	$\Delta_{\mathbf{SP}}\downarrow$	ACC(%)↑	F1(%)↑	$\Delta_{\mathbf{SP}}\downarrow$
NB	73.51	73.47	0.12	71.71±6.60	70.85±7.03	0.14±0.01
FairNB	73.02	72.95	0.10	71.27±5.22	70.50±5.65	0.05±0.05
CC-W	90.59	90.08	0.11	**91.66**±1.32	**91.59**±1.33	0.14±0.02
FairNB+CC-W	90.59	90.08	0.11	**91.66**±1.32	**91.59**±1.33	0.14±0.02
FairCC-ROC-W	90.59	90.06	0.08	89.08±1.87	88.95±1.85	0.05±0.04
FairCC-LF-W	89.60	89.04	0.05	89.78±1.07	89.69±1.11	0.04±0.00
FairCC-NR-W	87.13	86.68	**0.01**	88.59±0.96	88.47±0.98	**0.02**±0.01
FairCC-TA-W	**92.33**	**92.02**	0.09	90.37±0.92	90.27±0.91	0.07±0.03
CC-N	84.65	84.62	0.23	85.66±3.95	85.61±3.94	0.14±0.03
FairNB+CC-N	84.41	84.37	0.23	**87.20**±3.95	**87.12**±3.95	0.13±0.02
FairCC-ROC-N	**86.14**	**86.08**	0.23	85.61±4.07	85.55±4.06	0.06±0.01
FairCC-LF-N	81.68	81.64	0.05	85.26±3.23	85.21±3.22	0.04±0.00
FairCC-NR-N	78.96	78.92	**0.00**	84.37±3.93	84.34±3.95	**0.03**±0.02
FairCC-TA-N	82.43	82.39	0.06	78.01±9.54	77.81±9.67	0.07±0.04

1) Collective classification improves prediction performance for networked data: In Tables III and IV, the classic collective classification, either **CC-W** or **CC-N**, consistently achieves better prediction accuracy as well as F1-score than the **NB** classifier that makes predictions solely based on node features. This demonstrates the ability of collective classification models to utilize graph structure and properties in addition to node features to improve prediction performance. This gain can be especially attributed to the high degree of label-based homophily present in both German and Student network datasets which the collective classification model is able to use effectively to its advantage.

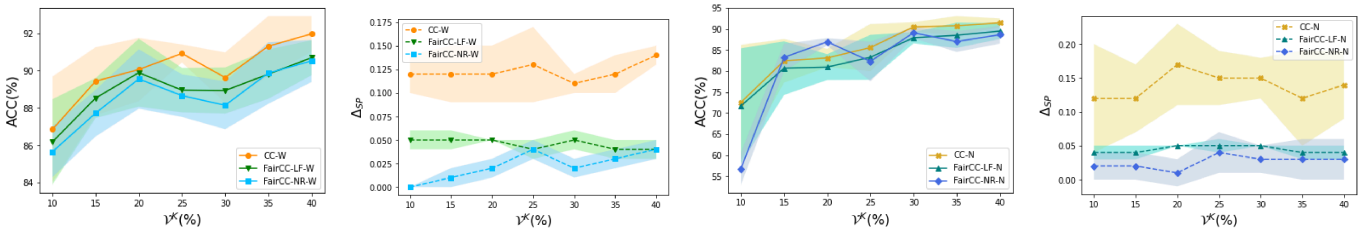
2) Collective classification cannot guarantee fair prediction: The unfairness measure Δ_{SP} is significant w.r.t. the widely-adopted threshold 0.05 for both **CC-W** and **CC-N** in the randomly sampled nodes as well as the degree-sorted sampled nodes for both datasets. Additionally, Δ_{SP} for **CC-W** is lower for degree-sorted sampling compared to random sampling for both datasets. In contrast, Δ_{SP} for **CC-N** with degree-sorted sampling is significantly larger than for **CC-N** with random sampling for the Student dataset. The opposite can be observed for the German dataset. This difference most likely stems from a higher degree of unfairness already present in $\mathcal{V}^K$ for the degree sampled Student dataset (0.089) compared to that of the German dataset (0.011). These observations suggest that the **CC-N** classifier is more sensitive to the composition of nodes in $\mathcal{V}^K$ and the amount of unfairness present in $\mathcal{V}^K$ compared to the **CC-W** classifier. As the **CC-N** classifier computes class prior and neighborhood class probability distribution using labels for $\mathcal{V}^K$ which are used to predict labels for $\mathcal{V}^U$ in every iteration, it is largely influenced by the neighborhood structure and bias present in $\mathcal{V}^K$.

3) The fair local classifier f_L is insufficient for fair collective classification: It is also evident from the results that the method of incorporating a fair local classifier into the collective classification framework, i.e., **FairNB+CC-W** or **FairNB+CC-N**, has significant Δ_{SP} values, indicating that a fair local classifier fails to ensure fair final predictions. Replacing **NB** with **FairNB** does not influence the final predictions for **CC-W** and only slightly influences **CC-N** predictions as the CC model uses **FairNB** initialization only once during the first iteration of inference while the consequent iterations in the classic rational classifiers may incur adverse bias. For the random sampling case, **FairNB+CC-N** can reduce Δ_{SP}, but



(a) ACC for f_R = WVRN (b) Δ_{SP} for f_R = WVRN (c) ACC for f_R = NBR (d) Δ_{SP} for f_R = NBR

Fig. 1: Effect of degree sampled $\mathcal{V}^k$ size on utility and fairness. (a): Accuracy (b): and statistical parity measures of **CC-W**, **FairCC-LF-W**, **FairCC-NR-W**, (c): Accuracy (d): and statistical parity measures of of **CC-N**, **FairCC-LF-N**, **FairCC-NR-N**. The Y-axis is not strictly set to 0 to focus on the trends.



(a) ACC for f_R = WVRN (b) Δ_{SP} for f_R = WVRN (c) ACC for f_R = NBR (d) Δ_{SP} for f_R = NBR

Fig. 2: Effect of randomly sampled $\mathcal{V}^k$ size on utility and fairness (a): Accuracy (b): and statistical parity measures of **CC-W**, **FairCC-LF-W**, **FairCC-NR-W**. (c): Accuracy (d): and statistical parity measures of of **CC-N**, **FairCC-LF-N**, **FairCC-NR-N**. The Y-axis is not strictly set to 0 to focus on the trends.

this decrease is not significant compared to other methods.

4) The proposed NR and TA mechanisms significantly mitigate bias in collective classification: The node reweighting (NR) method is able to achieve the specified fairness requirement (i.e., $\Delta_{SP} \leq 0.05$) for both WVRN and NBR classifiers under all settings. As opposed to the fair local classifier, FairCC with NR considers unfairness originating from known labels for $\mathcal{V}^K$ as well as predicted labels for $\mathcal{V}^U$ and mitigates bias iteratively in the relational classifiers. This approach can also significantly reduce the unfairness for both sampling techniques with a slight loss in accuracy. Despite this loss, **FairCC-NR-W** and **FairCC-NR-N** still maintain accuracy gain over the fair attribute-only classifier **FairNB**. This demonstrates the effectiveness of the node reweighting mechanism.

The second proposed mechanism, threshold adjustment (TA), significantly reduces unfairness, even though it fails to achieve the desired threshold $\delta = 0.05$ in some cases where the degree of unfairness is high in $\mathcal{V}^K$. Nonetheless, this method outperforms in the trade-off between accuracy and unfairness measure compared to the naive methods **FairNB+CC-W** and **FairNB+CC-N**.

5) The postprocessing mechanism reduces bias: The simple ranking-based label flipping approaches, **FairCC-LF-W** and **FairCC-LF-N**, achieve statistical parity under all settings. As the method is agnostic to the sampling technique and classification methodology, it is effective for both datasets under random sampling as well as degree-sorted sampling settings.

In contrast to the LF approach, the ROC postprocessing method is not agnostic to the sampling technique as it learns its region threshold parameter using probabilities for nodes in $\mathcal{V}^K$. Since the neighborhood class distribution of a node v_i influences v_i's label probability, the neighborhood class distribution in $\mathcal{V}^K$ influences the optimal parameter for ROC. For degree sampling, nodes in $\mathcal{V}^K$ have the highest degrees resulting in more populated and diverse 1-hop neighborhoods compared with the neighborhood for low degree nodes in $\mathcal{V}^U$. Hence the optimal region threshold learned for $\mathcal{V}^K$ may not be well suited to mitigate unfairness in $\mathcal{V}^U$. This is demonstrated in our results for both German and Student datasets. We further measure Δ_{SP} values for $\mathcal{V}^K$ and $\mathcal{V}^U$ after applying the ROC method which were observed to be 0.02, 0.08 for **FairCC-ROC-W** and 0.01, 0.23 for **FairCC-ROC-N** on the degree-sampled Student dataset. We observed similar differences on the German dataset. These results verify that the ROC mechanism is able to remove bias in $\mathcal{V}^K$ but fails to mitigate unfairness in $\mathcal{V}^U$ for degree-sorted sampling. Comparatively, the ROC method is able to reduce Δ_{SP} with a larger difference for random node sampling as the sampling method is likely to choose nodes with varying structural roles to form $\mathcal{V}^K$.

6) The proposed mechanisms consistently reduce bias with various $\mathcal{V}^K$ sizes: We further study the impacts of the size of $\mathcal{V}^K$ on NR and LF. Note that we skip the study of TA as it already fails to achieve fairness in some cases as shown in

Section V-C4 We conduct experiments on the Student dataset for both degree and random sampling as this dataset has more bias in $\mathcal{V}^K$. We vary $\mathcal{V}^K$ as $\{10\%, 15\%, 20\%, 25\%, 30\%, 35\%, 40\%\}$ of all nodes in the graph. The results for single runs of degree sampling are shown in Fig. 1a, 1b for CC and FairCC models with WVRN relational classifier and Fig. 1c, 1d for models with NBR relational classifier. Generally, Δ_{SP} is small when $\mathcal{V}^K$ is 10% but increases rapidly thereafter for vanilla CC. The proposed NR and LF can reduce unfairness significantly to achieve the specified fairness threshold with some drop in prediction performance for all tested $\mathcal{V}^K$ sizes. Fig. 2 shows the average results with standard deviation over 5 runs on randomly sampled Student dataset. The unfairness measure for vanilla CC is fairly similar despite differences in size of $\mathcal{V}^K$. Both NR and LF can consistently reduce unfairness in this setting as well thus validating the effectiveness of these methods.

VI. Conclusion

In this paper, we investigated collective classification from a fairness perspective and empirically verified that classic collective classification may result in unfair predictions w.r.t. the sensitive attribute. We investigated unfairness under two network-sampling techniques: random sampling and degree-sorted sampling for networked data exhibiting homophily. We formulated and tested various approaches for fair collective classification. We modified the collective classification framework to incorporate the node reweighting and threshold adjustment mechanisms for bias mitigation. We empirically verified the shortcomings of certain methodologies and the efficacy of the node reweighting and the postprocessing method with label flipping.

An interesting direction for future work would be to investigate fairness in more sophisticated forms of label propagation mechanisms such as the ones discussed in [22], [30]. Since the C&S framework can also be used as post-processing to improve GNN performance [22], it would be interesting to study the impact of fair propagation mechanisms on GNN predictions. Another direction could include extending FairCC formulations to include other popular measures of fairness such as equal opportunity [7], counterfactual fairness [15], individual fairness [6].

Reproducibility. All source code and datasets can be downloaded at https://tinyurl.com/4ck6kdca.

Acknowledgment

This work was supported in part by NSF 1946391, 2119691 and 2137335.

References

[1] S. A. Macskassy and F. J. Provost, "Classification in networked data: A toolkit and a univariate case study," *J. Mach. Learn. Res.*, pp. 935–983, 2007.

[2] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, pp. 93–106, 2008.

[3] J. Neville and D. D. Jensen, "Iterative classification in relational data," *AAAI Workshop on Learning Statistical Models from Relational Data*, pp. 13–20, 2000.

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

[5] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, pp. 115:1–115:35, 2021.

[6] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, "Fairness through awareness," in *Innovations in Theoretical Computer Science 2012*. ACM, 2012, pp. 214–226.

[7] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NeurIPS*, 2016, pp. 3315–3323.

[8] P. DiMaggio and F. Garip, "Network effects and social inequality," *Annual Review of Sociology*, pp. 93–118, 2012.

[9] C. O. Okafor, "Social networks as a mechanism for discrimination," *arXiv preprint arXiv:2006.15988*, 2020.

[10] G. Zeno and J. Neville, "Investigating the impact of graph structure and attribute correlation on collective classification performance," *MLG Workshop*, 2016.

[11] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowl. Inf. Syst.*, pp. 1–33, 2011.

[12] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *AISTATS*. PMLR, 2017, pp. 962–970.

[13] F. Kamiran, A. Karim, and X. Zhang, "Decision theory for discrimination-aware classification," in *ICDM*, 2012, pp. 924–929.

[14] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *ICLR*. PMLR, 2013, pp. 325–333.

[15] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *NeurIPS*, 2017.

[16] Y. Wu, L. Zhang, and X. Wu, "Counterfactual fairness: Unidentification, bound and algorithm," in *IJCAI*. ijcai.org, 2019, pp. 1438–1444.

[17] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. M. Wallach, "A reductions approach to fair classification," in *ICML*, 2018, pp. 60–69.

[18] G. Pleiss, M. Raghavan, F. Wu, J. M. Kleinberg, and K. Q. Weinberger, "On fairness and calibration," in *NeurIPS*, 2017, pp. 5680–5689.

[19] Y. Dong, J. Ma, C. Chen, and J. Li, "Fairness in graph mining: A survey," *arXiv preprint arXiv:2204.09888*, 2022.

[20] E. Dai *et al.*, "A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability," *arXiv preprint arXiv:2204.08570*, 2022.

[21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, pp. 4–24, 2021.

[22] Q. Huang, H. He, A. Singh, S. Lim, and A. R. Benson, "Combining label propagation and simple models out-performs graph neural networks," in *ICLR*, 2021.

[23] S. Tsioutsiouliklis, E. Pitoura, P. Tsaparas, I. Kleftakis, and N. Mamoulis, "Fairness-aware pagerank," in *WWW*, 2021, pp. 3815–3826.

[24] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. Networks*, pp. 107–117, 1998.

[25] L. Espín-Noboa, F. Karimi, B. Ribeiro, K. Lerman, and C. Wagner, "Explaining classification performance and bias via network structure and sampling technique," *Appl. Netw. Sci.*, p. 78, 2021.

[26] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," *ACM Sigmod Record*, pp. 307–318, 1998.

[27] C. Agarwal, H. Lakkaraju, and M. Zitnik, "Towards a unified framework for fair and stable graph representation learning," in *UAI*, 2021, pp. 2114–2124.

[28] T. Le Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi, "A survey on datasets for fairness-aware machine learning," *WIREs Data Mining and Knowledge Discovery*, p. e1452, 2022.

[29] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. T. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias," *IBM J. Res. Dev.*, pp. 4:1–4:15, 2019.

[30] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*. AAAI Press, 2003, pp. 912–919.