Fair Collective Classification in Networked Data

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

Yongkai Wu
Clemson University
Clemson, South Carolina, USA
yongkaw@clemson.edu

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

ABSTRACT

Collective classification utilizes network structure information via label propagation to improve prediction accuracy for node classification tasks. Because these models use information from labeled nodes which often contain historical bias, they may result in predictions that are biased w.r.t. the sensitive attributes of nodes. Throughout inference, this bias may even be amplified due to propagation. 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 (FairCC) framework and formulate various methodologies based on reweighting, threshold adjustment, and postprocessing to achieve fair prediction. Experiments on semi-synthetic datasets demonstrate the effectiveness of the proposed heuristics in significantly reducing prediction bias.

KEYWORDS

Collective classification, inference, label propagation, fairness.

ACM Reference Format:
Karuna Bhaila, Yongkai Wu, and Xintao Wu. 2022. Fair Collective Classification in Networked Data. In *Proceedings of 1st ACM SIGKDD Workshop on Ethical Artificial Intelligence: Methods and Applications (EAI-KDD).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Traditional machine learning models used for classification are often founded on the basis of an underlying assumption that the data is 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 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[13].

Collective classification (CC) utilizes network structure and underlying network properties such as label 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 and can be observed in real-life networks with varying strength. 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 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, researchers have been increasingly and rightfully concerned with making automated decision making systems socially and legally fair to prevent discrimination [6, 9]. Although, collective classification may improve prediction accuracy by utilizing homophily, it is imperative to analyze it through the lens of social inequality to ensure fair predictions for all demographic groups. DiMaggio et al. [3] studied social networks discrimination from a social science perspective and theorized that network effects can amplify discrimination in social networks by compounding initial endowments at the individual-level through normative influence. They suggest that such amplification occurs in homophilous networks where socio-economic characteristics are positively correlated with valuable resources, and an entity is influenced by its network peers to adopt similar practices. Initial endowment in social networks is analogous to historical bias found in most fair learning datasets. Furthermore, CC achieves better performance for networks with high label homophily[19]. It is thus possible for CC models to result in improved performance with seemingly accurate predictions that are biased against certain groups.

To address this issue, we study the problem of fair label propagation and inference. We empirically verify that CC can improve prediction accuracy but also result in unfair estimates. To mitigate the resulting bias, we develop a fair collective classification (FairCC) framework and incorporate various heurstic methodologies within it including an iterative reweighting method based on[10] and an iterative threshold adjustment method based on the covariance measure of unfairness[17], and postprocessing techniques [10, 11].

2 PRELIMINARIES

In this section, we describe our formulation for collective classification on an attributed graph. Throughout this paper, 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 $\mathbf{x}_{\cdot,j}$ to denote its i-th row and j-th column respectively. Due to space constraint, we include notation table, pseudo code for all algorithms, and discussion of related work in the Appendix.

2.1 Collective Classification

We extend the univariate formulation of collective classification [13] to include sensitive and non-sensitive 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 includes a sensitive attribute s_i. We use $\mathbf{x}_i \setminus s_i$ to denote its non-sensitive 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 $\mathcal{V}^U$. For simplicity, we assume both s and y to be binary.

The CC framework comprises three components: local classifier, relational classifier, and collective inference discussed as follows.

2.1.1 Local Classifier. The local classifier f_L estimates label probabilities $P(y_i|\mathbf{x}_i)$ using only node attributes $\mathbf{x}_i$. The classifier is trained with $\mathcal{V}^K$ to obtain initial labels/estimates for $\mathcal{V}^U$. The relational classifier uses these estimates to estimate probabilities for nodes whose neighborhood contains nodes from $\mathcal{V}^U$.

2.1.2 Relational Classifier. The relational classifier f_R leverages graph structure to directly estimate unknown node labels or a probability distribution over them. 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 $\mathcal{V} \setminus \mathcal{N}_i$ [13]. 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) [13] 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 edges.

Relational Bayes (NBR) also referred to as network-only Bayes [13] uses multinomial naive Bayesian classification based on the classes of v_i's neighbors.

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

$P(\mathcal{N}_i)$ is the same for all classes; normalization across classes allows us to avoid explicitly computing it. $P(y_i = c)$ is simply the class prior. Assuming independence between all neighbor classes of 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. Laplace smoothing is applied to account for possible zeros in the estimation of $P(y_j = \gamma|y_i = c)$.

2.1.3 Collective Inference. f_R 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 f_R. Then, 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.

Relaxation Labeling [13] is a collective inference method that performs an iterative update for the probability estimates of all $v_i \in \mathcal{V}_U$ at step $t+1$ based on estimations obtained from the f_R at step t. The update step is formulated as:

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

where, $\hat{\mathbf{c}}_i$ is a vector containing f_R's estimates of $P(y_i|\mathcal{N}_i)$, $\beta^0 = k$, $\beta^{(t+1)} = \alpha\beta^{(t)}$, k is a constant between 0 and 1, and α is a decay constant. We refer readers to Algorithm 1 in Appendix for an overall view of the CC framework.

2.2 Fairness Metrics

For evaluating bias in CC, we use statistical parity [6], a widely accepted notion of group fairness measured in terms of the binary sensitive attribute $s \in \{0,1\}$, and the binary predicted label $\hat{y} \in \{0,1\}$. Statistical parity requires the predictions be independent of the sensitive attribute, i.e., $\hat{y} \perp s$. It can be quantitatively evaluated as:

$$\Delta_{SP} = |P(\hat{y}|s = 0) - P(\hat{y}|s = 1)| \tag{6}$$

Other popular measures of fairness include equal opportunity [9], counterfactual fairness [12], individual fairness [6] whose exploration we leave for future work.

3 FAIR COLLECTIVE CLASSIFICATION

Fairness in CC can be evaluated at three different stages throughout the process: local fairness Δ_{f_L} for the predictions obtained from the f_L, relational fairness Δ_{f_R} for predictions obtained from f_R, and aggregated fairness Δ_C for the final predictions. Our goal ultimately is to ensure fairness guarantee in Δ_C.

One naive way of formulating FairCC is to ensure fairness in initial estimates for $\mathcal{V}^U$ computed by f_L. Since these fair initializations are propagated throughout the graph, it is reasonable to expect that imposing Δ_{f_L}-fairness in the f_L could help lead to overall fairness. However, a fair f_L cannot guarantee absolute fairness in $\mathcal{V}^U$. Moreover, the predictions from fair local classifier are used by the CC 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 4.

As bias amplification mainly occurs in f_R via propagation; ensuring Δ_{f_R}-fairness at every iteration of inference can lead to Δ_C-fairness. We propose two methodologies that make f_R predictions fair so that estimates obtained in every iteration are relatively fair.

3.1 Fair Collective Classification via Node Reweighting

The node reweighting (NR) method is based on the reweighting method [10] which aims to mitigate historical bias in data through preprocessing of the training data to make it fair and balanced. We propose to incorporate this reweighting technique in the CC model during each iterative update to assign weights to nodes w.r.t their s and y values. Instead of reweighing just the training set, we derive these weights for all nodes in the graph by using true labels for nodes in $\mathcal{V}^K$ and predicted labels in $\mathcal{V}^U$. This allows the computed weights to capture the historical bias in $\mathcal{V}^K$ as well as propagated bias in $\mathcal{V}^U$. The calculated weights are then propagated throughout the graph along with the inferred probabilities/labels.

Algorithm 2 in Appendix shows the pseudo-code of our FairCC via NR approach. We start by computing weights based on the known labels and predicted labels obtained from f_L.

$$w_{c,a}^{(0)} := \frac{P(y=c)}{P(y=c|s=a)} \tag{7}$$

Then for each iteration of inference, we derive weights using $\mathcal{V}^K$ labels, and $\mathcal{V}^U$ predictions from previous iteration until the predictions converge to a fair and stable state. Since the inference procedure incorporates simulated annealing in the update step to ensure label convergence[13], we perform a similar update for the weights by allowing the weights from the previous iteration to have the same amount of influence as the probability estimates from the previous iteration.

$$w_{c,a}^{(t+1)} = \beta^{(t+1)} \cdot \frac{P(y=c)^{(t)}}{P(y=c|s=a)^{(t)}} + (1-\beta^{(t+1)}) \cdot w_{c,a}^{(t)} \tag{8}$$

We leverage network homophily and apply these weights in the f_R such that positively labeled neighbor nodes from the privileged group have a smaller influence than negatively labeled neighbor nodes from the same group. The opposite effect is true for neighbor nodes from the unprivileged group. We further limit this influence by using the target node's own sensitive attribute-based weight. 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.

WVRN incorporates the computed weights by applying them directly to each neighbor's probability and also the node's own overall probability.

$$P(y_i = c|\mathcal{N}_i) = \frac{w_{c,s_i}}{Z} \sum_{v_j \in \mathcal{N}_i} P(y_j = c|\mathcal{N}_j) w_{c,s_j} \tag{9}$$

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.

$$P(y_i = c|\mathcal{N}_i) = P(y_i = c)\frac{1}{Z}\Big(\prod_{v_j \in \mathcal{N}_i} (P(y_j = \gamma | y_i = c))^{w_{\gamma,s_j}} \Big)^{w_{c,s_i}} \tag{10}$$

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 $1 \le a < b$ before applying them.

3.2 Fair Collective Classification via Threshold Adjustment

The threshold adjustment (TA) heuristic 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 [17], which quantifies the dependence between S and $\hat{Y}$ as a measure of covariance between the S and the signed distance of the node's feature vector from the decision boundary for linear classifiers. Minimizing this covariance measure then becomes analogous to minimizing the Δ_{SP}.

The decision boundary for CC models is analogous to 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 $\hat{\mathbf{c}}_{.,1}$ (assuming $y = 1$ to be the advantaged outcome) and the threshold value τ,

$$\text{Cov} = \frac{1}{n(\mathcal{V}^U)} \sum_{v_i \in \mathcal{V}^U} (s_i - \mu_{\mathrm{s}}) \cdot (\hat{c}_{i,1} - \tau) \tag{11}$$

where $n(\mathcal{V}^U)$ refers to the number of nodes in $\mathcal{V}^U$ and μ_{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 Eq. 11 as:

$$\text{Cov} = \frac{1}{n(\mathcal{V}^U)} \left[\sum_{v_i \in \mathcal{V}^U : s_i = +} (s_i - \mu_{\mathrm{s}}) \cdot (\hat{c}_{i,1} - \tau_+) + \sum_{v_i \in \mathcal{V}^U : s_i = -} (s_i - \mu_{\mathrm{s}}) \cdot (\hat{c}_{i,1} - \tau_-) \right] \tag{12}$$

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 Eq. 12 such that the computed covariance becomes 0.

$$\epsilon = \frac{-\text{Cov} \cdot n(\mathcal{V}^U)}{\sum_{v_i \in \mathcal{V}^U : s_i = +} (s_i - \mu_{\mathrm{s}}) - \sum_{v_i \in \mathcal{V}^U : s_i = -} (s_i - \mu_{\mathrm{s}})} \tag{13}$$

We compute ϵ and update τ_+ and τ_- in each iteration after the probability update step as shown in Algorithm 2 in Appendix. This allows us to iteratively approximate the optimal fair thresholds.

3.3 Fair Collective Classification via Post Processing

Another approach to reduce bias in CC is to directly reduce Δ_C-unfainess by modifying final predictions. To this end, we pair the CC framework with two postprocessing techniques as follows.

Reject Option Classification (ROC) [11] is a classifier-agnostic postprocessing method that exploits the low-confidence region of a probabilistic classifier for discrimination reduction. We adopt the same method and apply it over 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 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 is chosen according to label probabilities of nodes in $\mathcal{V}^K$ to maximize model performance and minimize bias.

Label Flipping (LF) [10] is another classifier-agnostic postprocessing method which directly changes the labels based on estimates from a probabilistic classifier. 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.

4 EXPERIMENTS

4.1 Compared Methods

We evaluate the fair collective classification formulations discussed in Section 3 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 WVRN and NBR relational models, respectively. We compare them against vanilla collective classification; CC-W and CC-N, and the naive methodology that uses a fair f_L; FairNB+CC-W and FairNB+CC-N. We also compare against non-collective classification methods, a Naive Bayes classifier, NB, and its fair version, FairNB, that implements exponentiated gradient reduction[1]. All methods evaluated in this section are implemented on the basis of AIF360 [2].

4.2 Results

Tables 1 summarizes our experiment results on the German dataset. Due to space constraint, we include details regarding datasets, and experimental setup as well as results for the Student dataset, and impact of $\mathcal{V}^K$ size on algorithm performance in the Appendix. We use accuracy and F1-score to evaluate performance and statistical parity to evaluate fairness of the models. From the tables, we draw the following conclusions.

4.2.1 Collective classification improves prediction performance for networked data. In Tables 1 and 4, the classic collective classification, either **CC-W** or **CC-N**, consistently achieves better prediction accuracy as well as F1-score than the attribute-only **NB** classifier. This demonstrates the ability of CC models to utilize graph structure and properties to improve prediction performance.

4.2.2 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** with random as well as degree-sorted sampling for both datasets. Additionally, Δ_{SP} values for the two sampling methods are fairly similar for **CC-W** compared to **CC-N**. For **CC-N**, Δ_{SP} is larger for degree-sorted sampling in the Student dataset and random sampling in the German dataset. We also observed a higher amount of bias in degree sampled $\mathcal{V}^K$ for Student dataset (0.089) than for the German dataset (0.011). These observations suggest that **CC-N** is more sensitive to the composition of nodes and bias in $\mathcal{V}^K$ compared to **CC-W**. As the **CC-N** computes class prior and neighborhood class probability distribution using nodes in $\mathcal{V}^K$, it is largely influenced by the neighborhood structure and bias present in $\mathcal{V}^K$.

4.2.3 The fair local classifier f_L is insufficient for fair collective classification. The method of incorporating a fair f_L into the CC framework, i.e., **FairNB+CC-W** or **FairNB+CC-N**, also has significant Δ_{SP} values, indicating that a fair f_L 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. For the random sampling case, **FairNB+CC-N** can reduce Δ_{SP}, but this decrease is not significant compared to other methods.

4.2.4 The proposed NR and TA mechanisms significantly mitigate bias in collective classification. The node reweighting (NR) method achieves the desired $\delta = 0.05$ for both WVRN and NBR classifiers under all settings. As opposed to using fair f_L in CC, FairCC-NR considers historical bias in $\mathcal{V}^K$ as well as propagated bias in $\mathcal{V}^U$ and mitigates it iteratively in subsequent iterations. This approach can also significantly reduce 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 **FairNB**. This demonstrates the effectiveness of FairCC-NR. The second proposed mechanism, threshold adjustment (TA), significantly reduces unfairness, even though it fails to achieve the desired $\delta = 0.05$ in some cases where the degree of unfairness is high in $\mathcal{V}^K$. In addition, this method outperforms in the trade-off between accuracy and unfairness measure than the naive methods **FairNB+CC-W** and **FairNB+CC-N**.

4.2.5 The postprocessing mechanism reduces bias. The simple ranking-based label flipping approach i.e., **FairCC-LF-W** and **FairCC-LF-N**, achieves 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. The ROC method i.e., **FairCC-ROC-W** and **FairCC-ROC-N**, reduces bias for both datasets with random sampling setting but fails to reduce bias with degree-sorted sampling. For degree-sorted sampling, nodes in $\mathcal{V}^K$ have the highest degrees resulting in more populated and diverse 1-hop neighborhoods compared with the low-degree nodes in $\mathcal{V}^U$; the region threshold θ obtained from $\mathcal{V}^K$ may not be optimal for $\mathcal{V}^U$, leading to a large bias for $\mathcal{V}^U$. We measure Δ_{SP} values for $\mathcal{V}^K$ and $\mathcal{V}^U$ which are observed to be 0.02, 0.08 for **FairCC-ROC-W** and 0.01,

Table 1: Results for German dataset

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

0.23 for **FairCC-ROC-N** on the degree-sampled Student dataset. We observed similar values for 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.

5 CONCLUSION

In this paper, we analyzed collective classification from a fairness perspective and empirically verified that CC can result in unfair predictions. We formulated and tested various approaches for fair collective classification. We modified the CC framework to incorporate reweighting and threshold adjustment mechanisms for bias mitigation. We empirically verified the shortcomings of certain methods and the efficacy of the reweighting and label flipping approaches.

Graph Neural Networks (GNNs) extend deep learning approaches to graph data and are equipped to handle large graphs[16]. Some studies have also extended their applications to heterogeneous graphs[4] and non-homophilous graphs [20]. Researchers have also proposed GNN frameworks that ensure fairness in graph mining tasks [5]. However, GNNs are computationally expensive and require large volume of data for training & tuning [16]. Therefore, for small graphs with favorable properties such as the ones discussed in this paper, fair collective classification methods still play an important role in node classification tasks. For future work, we will investigate bias in large-scale graphs with various structural properties. We will compare fair GNNs and fair CC, and explore their capabilities in various graph settings.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. 2018. A Reductions Approach to Fair Classification. In *ICML*. 60–69.
[2] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. *arXiv preprint arXiv:1810.01943* (2018). arXiv:1810.01943
[3] Paul DiMaggio and Filiz Garip. 2012. Network Effects and Social Inequality. *Annual Review of Sociology* (2012), 93–118.
[4] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *SIGKDD*. ACM, 135–144.
[5] Yushun Dong, Jing Ma, Chen Chen, and Jundong Li. 2022. Fairness in Graph Mining: A Survey. *arXiv preprint arXiv:2204.09888* (2022). arXiv:2204.09888
[6] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012*. ACM, 214–226.
[7] Lisette Espín-Noboa, Fariba Karimi, Bruno Ribeiro, Kristina Lerman, and Claudia Wagner. 2021. Explaining classification performance and bias via network structure and sampling technique. *Appl. Netw. Sci.* (2021), 78.
[8] Golnoosh Farnadi, Behrouz Babaki, and Lise Getoor. 2018. Fairness in Relational Domains. In *AIES*. ACM, 108–114.
[9] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NeurIPS*. 3315–3323.
[10] Faisal Kamiran and Toon Calders. 2011. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.* (2011), 1–33.
[11] Faisal Kamiran, Asim Karim, and Xiangliang Zhang. 2012. Decision Theory for Discrimination-Aware Classification. In *ICDM*. IEEE Computer Society, 924–929.
[12] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *NeurIPS*.
[13] Sofus A. Macskassy and Foster J. Provost. 2007. Classification in Networked Data: A Toolkit and a Univariate Case Study. *J. Mach. Learn. Res.* (2007), 935–983.
[14] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, and Eirini Ntoutsi. 2021. A survey on datasets for fairness-aware machine learning. *arXiv preprint arXiv:2110.00530* (2021).
[15] Yongkai Wu, Lu Zhang, and Xintao Wu. 2019. Counterfactual Fairness: Unidentification, Bound and Algorithm. In *IJCAI*. ijcai.org, 1438–1444.
[16] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* (2021), 4–24.
[17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *AISTATS*. PMLR, 962–970.
[18] Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. 2013. Learning Fair Representations. In *ICLR*. PMLR, 325–333.
[19] Giselle Zeno and Jennifer Neville. 2016. Investigating the impact of graph structure and attribute correlation on collective classification performance. *MLG Workshop* (2016).
[20] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. 2020. Beyond Homophily in Graph Neural Networks: Current Limitations and Effective Designs. In *NeurIPS*.

A RELATED WORK

A.1 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, 9], individual fairness which requires similar treatment or prediction for similar individuals [6, 18], and counterfactual fairness [12, 15] 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 [10, 18]. In-processing techniques involve a modified objective function that allows algorithm optimization subject to fairness constraints [1, 6, 17]. Post-processing directly changes the predicted labels to ensure fairness [9, 11]. However, most existing algorithms are applied under the assumption that data is iid which may not be directly applicable or effective for non-iid data and methodologies used in collective classification.

A.2 Fairness in Relational Learning

Some recent researches have explored the role of relational structures on prediction bias [8]. Farnadi et al. [8] introduced a notion of relational fairness that measures prediction bias generated from discriminative patterns in relational data. The proposed fair-PSL algorithm performs MAP inference subject to fairness constraints to predict values of unknown variables. Another work by Espin-Noboa et al. [7] studied the effects of network structure and sampling techniques on collective classification performance and bias. They compared the true positive rates for each label to assess the direction of bias. Their study concluded that bias in collective classification can be predicted based on class balance and level of homophily in the network, and the empirical results showed that minorities may be at a disadvantage when the classifier is trained on a small sample from homophilous or neutral networks with class imbalance.

The relational database setting described in [8] is different from our work where the dataset consists of a single table of attributed data accompanied by a network structure that connects entities described by those attributes. Our goal is to analyze and mitigate bias caused by label propagation, similar to the exploration in [7], rather than relational patterns among attributes. However, [7] performed collective classification on data 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, [7] did not propose or 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 non-sensitive 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.

B NOTATION

Table 2: 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
β	decay variable
$w_{c,a}$	the weight for a node with known/predicted label value c and sensitive attribute value a

C ALGORITHMS

Algorithm 1: Collective classification (CC)

```
Input  : $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X}, Y), f_L, f_R$
Output : $\{\hat{y}_i\}$
1 $\hat{\mathbf{c}}_i^{(0)} \leftarrow f_L(\mathbf{x}_i) \ \forall v_i \in \mathcal{V}^U$
2 for $t = 0 \ldots T$ do
3     $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \beta^{(t+1)} \cdot f_R(v_i)^{(t)} + (1-\beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)} \ \ \forall v_i \in \mathcal{V}^U$
4     use $\hat{\mathbf{c}}_i^{(t+1)}$ to obtain $\hat{y}_i^{(t+1)}$
5     if $\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)} \ \ \forall i \in \mathcal{V}^U$ then
6         break
7     end if
8 end for
9 return $\{\hat{y}_i^{(t+1)}\}$
```

D DATASETS

For the purpose of this study, we derive semi-synthetic datasets from two benchmarks for fair machine learning. 1) The German

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

```
Input  : $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X, Y), f_L, f_{wR}$
Output : fair $\{\hat{y}_i\}$
1  $\hat{\mathbf{c}}_i^{(0)} \leftarrow f_L(\mathbf{x}_i) \ \ \forall v_i \in \mathcal{V}^U$
2  $w_{c,a}^{(0)} \leftarrow 1 \ \ \forall c, \forall a$
3  for $t = 0 \dots T$ do
4      $w_{c,a}^{(t+1)} \leftarrow \beta^{(t+1)} \cdot P(y=c)^{(t)} / P(y=c|s=a)^{(t)}$
5      $+(1-\beta^{(t+1)}) \cdot w_{c,a}^{(t)} \ \ \forall c, \forall a$
6      $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \beta^{(t+1)} \cdot f_{wR}(v_i)^{(t)} + (1-\beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)} \ \ \forall v_i \in \mathcal{V}^U$
7      // $f_{wR}$ refers to Eq. 9 or 10
8      use $\hat{\mathbf{c}}_i^{(t+1)}$ to obtain $\hat{y}_i^{(t+1)}$
9      compute $\Delta_{SP}^{(t+1)}$
10     if $\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)} \ \ \forall i \in \mathcal{V}^U$ and $\Delta_{SP}^{(t+1)} \leq \delta$ then
11         break
12     end if
13 end for
14 return $\{\hat{y}_i^{(t+1)}\}$
```

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

```
Input  : $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X, Y), f_L, f_R$
Output : fair $\{\hat{y}_i\}$
1  $\hat{\mathbf{c}}_i^{(0)} \leftarrow f_L(\mathbf{x}_i) \ \ \forall v_i \in \mathcal{V}^U$
2  Initialize $\tau_+ = \tau_- = 0.5$
3  for $t = 0 \dots T$ do
4      $\hat{\mathbf{c}}_i^{(t+1)} \leftarrow \beta^{(t+1)} \cdot f_R(v_i)^{(t)} + (1-\beta^{(t+1)}) \cdot \hat{\mathbf{c}}_i^{(t)} \ \ \forall v_i \in \mathcal{V}^U$
5      compute $\epsilon^{(t+1)}$
6      // Refer to Section 3.2
7      $\tau_+ \leftarrow \tau_+ + \beta^{(t+1)} \cdot \epsilon^{(t+1)} + (1-\beta^{(t+1)}) \cdot \epsilon^{(t)}$
8      $\tau_- \leftarrow \tau_- - \beta^{(t+1)} \cdot \epsilon^{(t+1)} - (1-\beta^{(t+1)}) \cdot \epsilon^{(t)}$
9      use $\hat{\mathbf{c}}_i^{(t+1)}$ and $\tau_+$ to obtain $\hat{y}_i^{(t+1)} \ \ \forall i \in \mathcal{V}^U : s_i = +$
10     use $\hat{\mathbf{c}}_i^{(t+1)}$ and $\tau_-$ to obtain $\hat{y}_i^{(t+1)} \ \ \forall i \in \mathcal{V}^U : s_i = -$
11     compute $\Delta_{SP}^{(t+1)}$
12     if $\hat{y}_i^{(t+1)} = \hat{y}_i^{(t)} \ \ \forall i \in \mathcal{V}^U$ and $\Delta_{SP}^{(t+1)} \leq \delta$ then
13         break
14     end if
15 end for
16 return $\{\hat{y}_i^{(t+1)}\}$
```

credit dataset [14] 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 [14] describes student achievements in Portuguese subject at two Portugese 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 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 3: Dataset Statistics

Dataset	German	Student
# of nodes	955	577
# of edges	19980	8411
# of node attributes	32	38
density	0.04	0.05
assortativity in y	0.667	0.649
assortativity in s	0.595	0.573

Table 3 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.

E EXPERIMENTAL SETUP

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 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 $\mathcal{V}^K$.

We use $\delta = 0.05$ as the fairness threshold in the experiments. Following the experimental setup in [13], we set $k = 1$, $\alpha = 0.99$ and run the inference procedure for a maximum of 100 iterations for both CC and FairCC. 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.

F MORE EXPERIMENTAL RESULTS

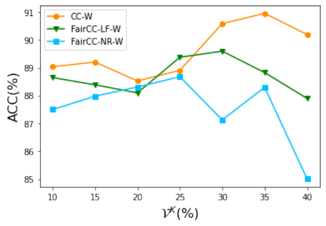
F.1 Results for Student dataset

F.2 Impact of $\mathcal{V}^K$ size

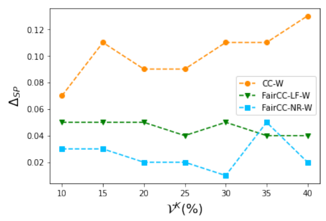
We further study the impacts of the size of $\mathcal{V}^K$ on NR and LF. Note that we skip the study of TA as it already fails to achieve fairness in some cases as shown in Section 4.2.4 We conduct experiments on the Student dataset for both degree and random sampling as this dataset has more bias in $\mathcal{V}^K$. 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 Figures 1a, 1b for CC and

Table 4: Results for Student dataset

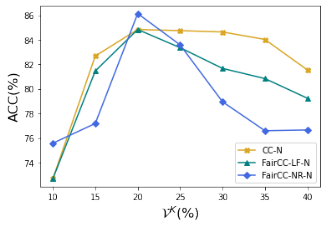
Method	Degree			Random		
	ACC(%)↑	F1(%)↑	Δ_{SP} ↓	ACC(%)↑	F1(%)↑	Δ_{SP} ↓
NB	73.51	73.47	0.12	71.71$\pm$6.60	70.85$\pm$7.03	0.14$\pm$0.01
FairNB	73.02	72.95	0.10	71.27$\pm$5.22	70.50$\pm$5.65	0.05$\pm$0.05
CC-W	90.59	90.08	0.11	**91.66**$\pm$1.32	**91.59**$\pm$1.33	0.14$\pm$0.02
FairNB+CC-W	90.59	90.08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



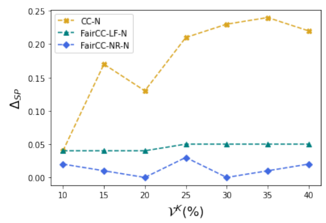
(a) ACC for f_R = WVRN



(b) Δ_{SP} for f_R = WVRN

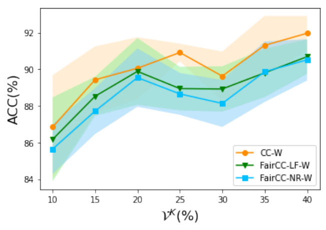


(c) ACC for f_R = NBR

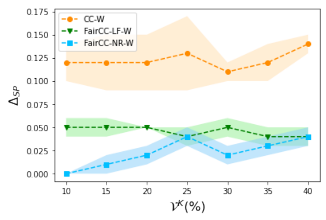


(d) Δ_{SP} for f_R = NBR

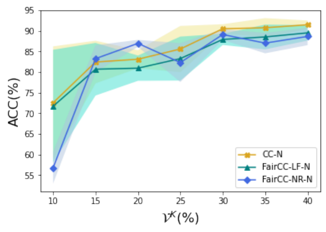
Figure 1: (a): The accuracy measures (b): and statistical parity measures of CC-W, FairCC-LF-W, FairCC-NR-W under different degree-sorted Student $\mathcal{V}^K$ size. (c): The accuracy measures (d): and statistical parity measures of of CC-N, FairCC-LF-N, FairCC-NR-N under different degree-sorted Student $\mathcal{V}^K$ size.



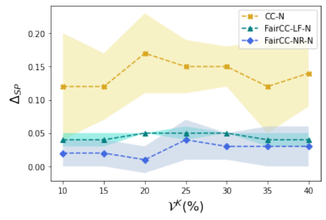
(a) ACC for f_R = WVRN



(b) Δ_{SP} for f_R = WVRN



(c) ACC for f_R = NBR



(d) Δ_{SP} for f_R = NBR

Figure 2: (a): The accuracy measures (b): and statistical parity measures of CC-W, FairCC-LF-W, FairCC-NR-W under different randomly Student $\mathcal{V}^K$ size. (c): The accuracy measures (d): and statistical parity measures of of CC-N, FairCC-LF-N, FairCC-NR-N under different randomly Student $\mathcal{V}^K$ size.

FairCC models with WVRN relational classifier and Figures 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. Figure 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. These results demonstrate that both NR and LF can effectively achieve fairness for various sizes of $\mathcal{V}^K$ with a slight loss in accuracy.