Edge Importance Scores for Editing Graph Topology to Preserve Fairness

Sree Harsha Tanneru [*1]

Abstract

Graph neural networks have shown promising performance on graph analytical tasks such as node classification and link prediction, contributing to great advances in many graph-based applications. Despite the success of graph neural networks, most of them lack fairness considerations. Consequently, they could yield discriminatory results towards certain populations when such algorithms are exploited in high stakes applications. In this work, we study the problem of predictive bias propagated by relational information, and subsequently propose an in-training edge editing approach to promote fairness. We introduce the notions of faithfulness and unfairness for an edge in a graph, and use it as prior knowledge to edit graph topology and improve fairness.

1. Introduction

Discriminatory bias can appear in many human-centered applications of graph neural networks, where data has been historically generated unfairly. As a result, predictive models built on this data have continued to perpetuate these biases while failing to recognize the historical context of the data. Most social networks are observed to be homophily-dominant. Nodes in the local neighbourhood belong to the same sensitive class with minimal connections across nodes of differing sensitive attributes. Therefore communities isolate themselves polarizing the opinions expressed within the communities. In this work, we study to problem of predictive bias propagated by relational information in node classification. We also propose an in-training approach to edit graph topology to mitigate bias.

[*]Equal contribution [1]Institute for Applied Computation, Harvard University, Cambridge, USA. Correspondence to: Sree Harsha Tanneru <sreeharshatanneru@g.harvard.edu>.

Presented at the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023.

1.1. Related Work

Fairness in Machine Learning

In the past decade, researchers have proposed a variety of fairness definitions, each with its own strengths and weaknesses. The notion of Individual Fairness (Dwork et al., 2011) and Disparate Treatment (Zafar et al., 2017) emphasizes that individuals from different sensitive groups should have similar outcomes if they have similar non-sensitive attributes. Counterfactual Fairness (Kusner et al., 2018) captures the intuition that a decision is fair if it is the same in the actual world and in the counterfactual world where the individual belong to a different sensitive group. In contrast, Group Fairness concentrates on statistical parity, including demographic parity (or disparate impact) (Zafar et al., 2017), and equality of opportunity (Hardt et al., 2016). While previous notions focused on prediction outcomes, Fairness Through Unawareness (Grgic-Hlaca et al., 2016) focused on process fairness, which requires that the sensitive feature are not explicitly included in the decision-making process. More recent works have introduced the notions of faithfulness (or fidelity) (Zhou et al., 2021) (Liu et al., 2021) (Hooker et al., 2018), stability (or robustness) (Alvarez-Melis & Jaakkola, 2018), and proposed metrics to quantify these notions.

Fairness in Graph Representation Learning

As previously mentioned, a major source of bias in graph representation learning is homophily, which means that similar nodes in graphs tend to interact with each other. Several research works explore edge rewiring and edge rebalancing approaches to counter possible bias that could arise from homophily. FairDrop (Spinelli et al., 2022) proposes to create a random copy of the adjacency matrix biased towards a decrease in homophily and reduce predictability of its sensitive attributes. Nifty (Agarwal et al., 2021) tries to adopt an adversarial-like training paradigm that perturbs the graph towards a more fair objective. However, this work fails to consider adding an edge when perturbing the graph's structure. FairGNN (Dai & Wang, 2021) considers non-i.i.d. data and tries to leverage graph structure with limited sensitive information, while maintaining high computation efficiency. FairEdit (Loveland et al., 2022) considers debiasing the input graph during training with the addition of

artificial nodes and edges and not just the deletion. However, FairEdit (Loveland et al., 2022) samples edges uniformly but each edge might contribute differently to model accuracy and group fairness. Our work incorporates fairness considerations for more selective edge sampling. Moreover, FairEdit (Loveland et al., 2022) edits only one edge at every epoch. Real-world graphs have thousands of nodes and edges and editing one edge is not usually enough to make a meaningful impact on group fairness.

2. Background and Notation

2.1. Notation

In our study, we focus on node classification with binary sensitive attribution. We denote $\hat{Y}$ and S as random variables that represent the predicted class label and sensitive feature, respectively, for a randomly selected node in the input graph. The sensitive feature is binary and divides the population into two subgroups based on a characteristic such as sex (e.g., male/female).

2.2. Group Fairness

Group fairness necessitates that the algorithm does not produce discriminatory predictions or decisions against individuals belonging to any particular sensitive subgroup. In this section, we present two popular notions of group fairness - demographic parity and equal opportunity.

Demographic Parity Demographic parity is attained when the model exhibits an equal acceptance rate for individuals in both sensitive subgroups. The extent of demographic parity is quantified by Δ_{DP}.

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

Equal opportunity Equality of opportunity asserts that positive predictions should be made independent of sensitive features for individuals with positive ground truth labels.

$$\Delta_{EO} = |P(\hat{Y} = 1|S = 0, Y = 1) - P(\hat{Y} = 1|S = 1, Y = 1)|$$

3. Approach

3.1. Fairness through Edge Editing

Biases observed in graph neural networks often stem from the biased network topology. Edge rewiring is a commonly employed debiasing strategy that helps mitigate bias resulting from homophily i.e; the tendency of nodes in a graph to connect with similar nodes. The key assumptions behind edge-rewiring is that graph data generated through discriminatory means is either a) missing edges that would have been present in more fair settings, or b) over-representing homophilous edges due to social stratification. Figure 1 illustrates a basic pipeline of edge rewiring. While several works study perturbing graph topology to improve fairness, none of them explicitly incorporate fairness considerations to create perturbations.

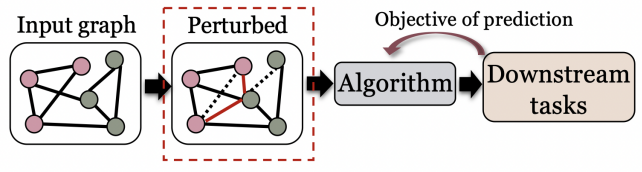


Figure 1. Edge Editing pipeline. Input graph topology is edited (i.e; drop existing edges, add new edges) to obtain fair algorithm output in downstream tasks. Figure by (Dong et al., 2023)

3.2. Faithfulness and Unfairness of Edges

We introduce the notions of faithfulness and unfairness for each edge in a graph.

1. **Unfairness Score** b_{e_i}: defines the contribution of edge towards the group fairness in the graph, measured as the gradient of demographic parity Δ_{DP} $b_{e_i} = |\frac{\partial \Delta_{DP}}{\partial e_i}|$.

2. **Faithfulness Score** f_{e_i}: defines the contribution of edge to model predictions, measured as the gradient of utility loss $\mathcal{L}_{\text{utility}}$ $f_{e_i} = |\frac{\partial \mathcal{L}_{\text{utility}}}{\partial e_i}|$.

3.3. Algorithm

The algorithm involves proposing a set of counterfactual edges E^* to be edited by (i) Adding edges between nodes with different sensitive attributes with a probability ρ (ii)Removing edges between nodes with the same sensitive attributes with a probability γ, thereby creating a new perturbed graph G^*. For the perturbated graph, faithfulness and unfairness scores for each edge are computed using backward propagation of gradients with respect to $\mathcal{L}_{\text{utility}}$ and Δ_{DP}. After assigning to every edge in the graph, we drop the top k unfair edges which are not faithful (i.e; $f_{e_i} <$ threshold) to model predictions in each epoch. The pseudo code for the algorithm is given in 1.

Algorithm 1 Editing edges for fair training of graph neural networks

Input : Training Graph $G = (V, E, X)$, sensitive attribute s_i, # of epochs to perturb graph α, # of training epochs K, Optimizer O
Output : Optimized model parameters θ
for $k \leftarrow 1$ *to* K **do**
  Train one step using optimizer O for model parameters θ on graph G
  if $k < \alpha$ **then**
    Generate perturbed graph G^* wuth edges $E*$
    Calculate f_{e_i} and b_{e_i} for each edge $e_i \in E$
    Drop top-K unfair edges which are not faithful
    $G \leftarrow G^*$
  end
end
return θ

4. Results

4.1. Credit Default, German Credit and Recidivism Datasets

We use three datasets proposed by (Agarwal et al., 2021) in order to evaluate graph neural networks on node classification. These data sets have a well defined sensitive attribute that can be probed for fairness. **Recidivism** dataset has nodes representing a defendant who was released on bail in the U.S state court system during 1990-2009. Connections are related to similarity of crimes and past convictions. The classification task is to determine whether a defendant would receive bail and uses race as the sensitive attribute. **Credit Default** dataset has nodes representing individuals who are utilizing some form of credit. Individuals are connected by their spending and payment behavior. The classification task is to determine whether an individual will default on the credit card payment. Age is the sensitive attribute. **German Credit** dataset has nodes representing an individual who uses a specific German bank. Each connection identifies a similarity in credit account history. The classification task is to classify individuals into those who have high and low credit risk. The individual's gender is the sensitive attribute.

We compare our edge editing approach with two baselines. First baseline is fairness through regularisation, where we add an extra term to the optimization objective to promote the fairness level of the algorithm output. ($\mathcal{L} = \mathcal{L}_{\text{utility}} + \lambda\ \mathcal{L}_{\text{fair}}$, $\mathcal{L}_{\text{fair}} = \Delta_{DP}$). The second baseline is fairness through unawareness (*FTU*) where sensitive features are not explicitly included in the decision-making process. This is often seen as a way to address the problem of bias in machine learning models, as it prevents the models from being influenced by factors that should not be considered in making decisions. However, there is a lot of critique that *FTU* simply ignores the problem of bias, rather than addressing it head-on.

We conducted experiments on three graph neural network architectures—GCN (Graph Convolution Network), GraphSAGE (Sample and aggregate), and APPNP (personalized propagation of neural predictions). Each of these architectures have different message passing mechanisms, and hence acts as test if the fairness improvement approach is model agnostic. The results are shown in table 1.

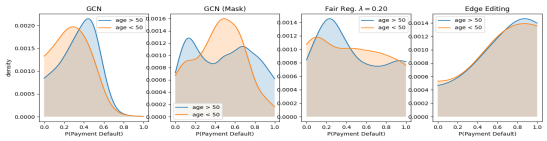


Figure 2. Outcome distribution of different approaches on Credit Default dataset

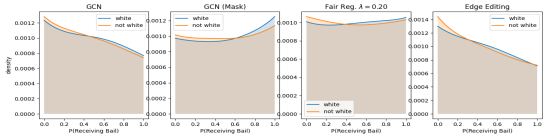


Figure 3. Outcome distribution of different approaches on Recidivism dataset

In our study, we found that although the sensitive attribute was masked, significant demographic parity still existed, indicating the presence of bias in other node attributes and relational information. As the regularization factor λ increases, parity improved, but at the cost of a significant drop in model utility, as measured by F_1 score and accuracy. On the other hand, edge editing improved parity without compromising accuracy. Furthermore, the outcome distributions shown for credit default, german credit, and recidivism datasets (shown in figures 2, 6, and 3) are much closer to each other in edge editing, compared to regularization and sensitive attribute masking. Finally, the observed trends in fairness and accuracy are consistent across all three architectures indicating that the algorithm is model agnostic.

4.2. Synthetic Data

In addition to these three real-world datasets, we run experiments on a synthetic dataset. We considered a binary sensitive attribute setting, where data is independently and identically sampled from two Gaussian distributions where each distribution represents an output class. To be consistent with our group fairness goals (i.e., sensitive attributes should not have any bearing on outcome), we assigned labels uniformly at random to nodes from both clusters. Furthermore, we added relational information as follows (i) Add edges between nodes sampled from the same distribution with

Dataset	Credit Default			German Credit			Recidivism		
Model	**F1**	Δ_{SP}	Δ_{EO}	**F1**	Δ_{SP}	Δ_{EO}	**F1**	Δ_{SP}	Δ_{EO}
GCN	81.7	7.56	6.47	80.3	5.77	6.72	78.7	10.3	8.09
SAGE	81.8	12.1	9.82	80.5	9.14	7.35	82.9	0.56	0.03
APP NP	81.4	14.7	13.0	78.6	30.6	29.9	78.7	6.85	6.53
GCN (Mask)	87.2	0.46	0.09	82.2	14.9	10.7	79.8	0.55	2.31
SAGE (Mask)	85.1	14.1	11.6	77.3	33.1	24.1	81.5	6.29	5.45
APP NP (Mask)	83.7	11.5	9.81	76.8	36.7	29.6	76.3	4.47	3.42
GCN + Fair Reg. ($\lambda = 0.1$)	79.0	2.37	0.12	77.2	9.11	9.56	81.4	7.51	5.26
SAGE + Fair Reg. ($\lambda = 0.1$)	82.1	7.15	5.32	81.6	0.23	2.84	81.0	5.54	3.62
APP NP + Fair Reg. ($\lambda = 0.1$)	81.4	4.00	3.17	63.4	5.33	1.89	77.1	6.23	5.05
GCN + Fair Reg. ($\lambda = 0.2$)	85.0	10.7	11.2	63.3	3.19	5.04	79.3	8.38	5.63
SAGE + Fair Reg. ($\lambda = 0.2$)	81.7	7.47	5.21	67.9	6.42	1.47	81.5	7.39	6.69
APP NP + Fair Reg. ($\lambda = 0.2$)	82.1	4.99	3.32	61.8	4.48	8.51	75.4	5.31	4.10
GCN + Edge Editing	**85.7**	**3.08**	**3.16**	**80.9**	**3.21**	**4.21**	**79.3**	**7.98**	**4.76**
APP NP + Edge Editing	**87.6**	**0.02**	**0.02**	**81.4**	**2.58**	**0.63**	**80.2**	**5.70**	**4.84**

Table 1. Accuracy, F_1, Δ_{DP}, and Δ_{EO} metrics on Recidivism, Credit Default, and German Credit datasets. All numbers are percentages.

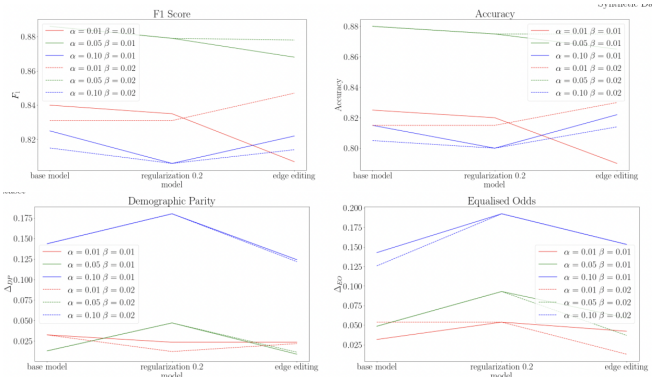


Figure 4. GCN performance at different α and β

probability α and different distributions with probability β. We set $\beta \leq \alpha$ as we are interested in homophilous graphs. An illustration is shown in 5.

We vary α and β and measure classification performance and group fairness of a graph convolutional network. This way we can better understand what happens to performance and fairness as graph becomes more homophilous. As α (intra-cluster similarity) increases, we notice that edge editing outperforms regularization and the base model. This proves that our method works as expected. However, we notice a few anomalies in the results, notably when $\alpha = \beta$, edge editing results in a drop in model accuracy and f1 scores. Moreover, at high values of α, optimization with regularization yields higher parity. We hypothesize that this inconsistencies could be due to bad hyper-parameters.

Discussion and Conclusion

We empirically evaluate group fairness of outcomes of standard graph neural network architectures on three real-world datasets and a synthetic dataset. As a baseline approach for improving group fairness, we try regularizing the loss with demographic parity. However, the improvement in parity comes with a drop in model utility. We then explore edge rewiring methods for improving group fairness, however most approaches fails to consider that not every edge has equal importance to accuracy and fairness in a graph. We introduce the notions of faithfulness and unfairness for edges and use this information to drop edges in training. We demonstrate an improvement in group fairness without drop in model utility. However, there are drawbacks of optimizing for Δ_{DP}, that drastically different output distributions can have negligible demographic parity. This is because Δ_{DP} only considers the difference in mean output scores across subgroups. In such scenarios, optimizing a better distance metric for probability distributions like KL Divergence (D_{KL}) would be more appropriate.

Furthermore, despite previous works showing editing edges improves adversarial robustness (Zügner et al., 2018), there are often claims that adding fictitious links (Li et al., 2021) and dropping existing links selectively might alter message passing and corrupt representation learning. Empirically or theoretically, analysing the extent to which a graph topology can be changed without corrupting representations is an exciting future line of research.

References

Agarwal, C., Lakkaraju, H., and Zitnik, M. Towards a unified framework for fair and stable graph representation

learning, 2021.

Alvarez-Melis, D. and Jaakkola, T. S. On the robustness of interpretability methods, 2018.

Dai, E. and Wang, S. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information, 2021.

Dong, Y., Ma, J., Wang, S., Chen, C., and Li, J. Fairness in graph mining: A survey, 2023.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness, 2011.

Grgic-Hlaca, N., Zafar, M. B., Gummadi, K. P., and Weller, A. The case for process fairness in learning: Feature selection for fair decision making. 2016.

Hardt, M., Price, E., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

Hooker, S., Erhan, D., jan Kindermans, P., and Kim, B. Evaluating feature importance estimates. *arXiv*, 2018. URL https://arxiv.org/pdf/1806.10758.pdf.

Kusner, M. J., Loftus, J. R., Russell, C., and Silva, R. Counterfactual fairness, 2018.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, P., Wang, Y., Zhao, H., Hong, P., and Liu, H. On dyadic fairness: Exploring and mitigating bias in graph connections. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=xgGS6PmzNq6.

Liu, Y., Khandagale, S., White, C., and Neiswanger, W. Synthetic benchmarks for scientific research in explainable machine learning, 2021.

Loveland, D., Pan, J., Bhathena, A. F., and Lu, Y. Fairedit: Preserving fairness in graph neural networks through greedy graph editing, 2022.

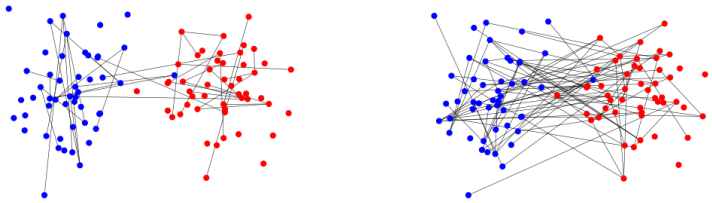
Spinelli, I., Scardapane, S., Hussain, A., and Uncini, A. FairDrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Transactions on Artificial Intelligence*, 3(3):344–354, jun 2022. doi: 10.1109/tai.2021.3133818. URL https://doi.org/10.1109%2Ftai.2021.3133818.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. Fairness Constraints: Mechanisms for Fair Classification. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 962–970. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/zafar17a.html.

Zhou, J., Gandomi, A. H., Chen, F., and Holzinger, A. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5), 2021. ISSN 2079-9292. doi: 10.3390/electronics10050593. URL https://www.mdpi.com/2079-9292/10/5/593.

Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp Data Mining*. ACM, jul 2018. doi: 10.1145/3219819.3220078. URL https://doi.org/10.1145%2F3219819.3220078.

A. Appendix



(a) $\alpha = \beta$ (b) $\alpha < \beta$

Figure 5. Illustration of synthetic dataset

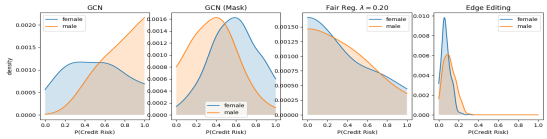


Figure 6. Outcome distribution of different approaches on German Credit dataset