ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Biased Edge Dropout in NIFTY for Fair Graph Representation Learning

Federico Caldart[1], Luca Pasa[1], Luca Oneto[2], Alessandro Sperduti[1], and Nicolò Navarin[1,*]

1 - University of Padua, Via Trieste 63, 35121, Padua, Italy

2 - University of Genoa, Via Opera Pia 11a, 16145, Genova, Italy

Abstract. Graph Neural Networks (GNNs) are nowadays widely used in many real-world applications. Nonetheless, the data relationships can be a source of biases based on sensitive attributes (e.g., gender or ethnicity). Several methods have been proposed to learn fair graph node representations. In this work we extend NIFTY, an approach that exploits additional terms in the loss function based on perturbing the input data to enforce the fairness of the GNNs. In particular, we exploit a biased perturbation of the adjacency matrix of the graph able to reduce the edge homophily. We show the effectiveness of our approach in four real-world graph datasets.

1 Introduction

Recent developments in Machine Learning (ML) systems made it possible to achieve or even surpass human performance in many real-life tasks. These achievements are now possible thanks to the availability of huge amounts of data that are used to train such ML models. Unfortunately, these high-performance level allow these models to also inherit human biases hidden in the data. When the decisions of these models may affect people's life we cannot allow these biased behaviours. In particular, we would like our model to be *fair*, namely to treat equally different subgroups of the population based on characteristics such as gender or ethnicity, referred to as *sensitive* attributes. The problem is even more challenging when the input data is complex (e.g., graphs) and black-box models such as Graph Neural Networks (GNNs) need to be employed to achieve satisfactory performance. Different methods have been proposed in literature to learn fair graph node representations, mainly including additional terms in the loss function to account for some definition of fairness [1, 2], or perturbing the graph topology [3, 4] in a biased way. As a representative e of the first family of methods, NIFTY [2] proposes that some of the additional terms can be based on perturbing the input data such as node attributes (including the sensitive attribute) or graph topology [2]. The latter methods are based on the intuition that GNNs tend to smooth the learned representations of connected nodes, and if the graph shows high homophily with respect to the sensitive attribute (i.e., nodes with the same value for the sensitive attribute tend to be connected), the GNN may introduce inequalities in the learned representations.

In this paper, we propose to unify the two families of approaches. We start from NIFTY and replace the random perturbations of the adjacency matrix with a biased perturbation reducing edge homophily. Experimental results on

*This work was partly funded by the SID/BIRD project *Deep Graph Memory Networks*, Department of Mathematics, University of Padua.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

four real-world datasets show that our proposal can improve different fairness metrics compared to the original NIFTY formulation, while maintaining the same computational complexity and the same level of predictive performance.

2 Background

Let us consider a graph $G = \{V, E, X, \mathbf{s}\}$ where $V = \{v_0, \ldots, v_{n-1}\}$ is the set of nodes or vertices, $E \subseteq V \times V$ is the set of edges, $X \in \mathbb{R}^{n \times d}$ is the matrix of non-sensitive node features, and $\mathbf{s} \in \{0,1\}^n$ is the vector associating a value for the binary (for sake of simplicity) sensitive feature to each node. We define $A \in \mathbb{R}^{n \times n}$ as the adjacency matrix of the graph.With $\mathcal{N}(v)$ we denote the set of nodes adjacent to node v. Let also $\tilde{D} \in \mathbb{R}^{n \times n}$ be the diagonal degree matrix where $d_{ii} = \sum_j a_{ij}$.

A GNN is a model that exploits the structure of the graph and the information embedded in feature vectors of each node in order to learn a representation $\mathbf{h}_v \in \mathbb{R}^m$ for each vertex $v \in V$. The first works extending neural networks to inputs in the graph domain [5, 6, 7] are based on the idea of aggregating the representation of a node and its neighbors, either in a recursive or a feed-forward (convolutive) way. This idea has been re-branded later as *graph convolution*. In the last few years, several different GCs have been proposed. In this work we build on top of a widely adopted graph convolutions, dubbed GCN [8], that is defined as $H = \dot{D}^{-\frac{1}{2}} \dot{A} \dot{D}^{-\frac{1}{2}} X\Theta$, where $\dot{A} = A + I$ and $\dot{d}_{i,j} = \sum_{j=0}^{n} \dot{a}_{ij}$. Usually, GNNs stack multiple Graph Convolutional layers to consider wider topological receptive fields.

The task we consider in this paper is node classification. Moreover, we would like our learned hypothesis h to be *fair* according to some metric. A common fairness metric is *Statistical (or Demographic) Parity* (SP), defined as: $\Delta_{SP} = P(\hat{y} = 1|s = 0) - P(\hat{y} = 1|, s = 1)$, where $\hat{y}$ indicates the predicted outcome for a node (i.e., the output of the function h applied to the node). Another common metric is *Equal Opportunity* (EO) [9], defined as: $\Delta_{EO} = P(\hat{y} = 1|y = 1, s = 0) - P(\hat{y} = 1|y = 1, s = 1)$. The probabilities in both metrics are usually estimated on the validation or test sets.

3 Related Works

In this section, we discuss the most relevant methods to enforce fairness on tasks defined over graphs. The two main approaches to learn fair representations are: *(i)* to define models that learn fair representations from biased input data and *(ii)* methods that modify the input in order to make it less biased. For the first approach, several proposed methods exploits adversarial learning [10, 11], probabilistic approaches [12, 13], or specifically designed loss functions [14, 1].

In this work, we will exploit NIFTY (uNIfying Fairness and stabiliTY) [2] that aims to learn both fair and stable node representations. The method generates several augmented versions of the input graph, in which the original node attributes and edges are slightly perturbed, obtained by: *(i)* perturbing node attributes as $\tilde{x}_v = x_v + r \circ \alpha$, where $r \in \{0,1\}^M$ is a random masking vector drawn from a Bernoulli distribution and α is sampled from a normal distribution,

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

(ii) perturbing the sensitive attribute, modifying it to generate a counterfactual, i.e. by flipping the sensitive attribute s in x_v, and *(iii)* exploiting a perturbed adjacency matrix $\tilde{A} = A \circ R_e$ where $R_e \in \{0,1\}^M$ is a random mask sample from a Bernoulli distribution. The random mask removes some graph edges, similarly to what happens by using Drop-Edge [15]. The proposed objective function maximizes the similarity between the learned embedding of the original graph nodes and their counterparts in the augmented graph, relying on a siamese GNN encoder that generates the representations of each graph node $\tilde{z}_v$ and its augmented version $\bar{z}_v$. Then a predictor $t : \mathbb{R}^d \to \mathbb{R}^d$ is used to transform and match corresponding representations. The training minimizes the following loss function: $L = \min_{\Theta} \mathbb{E}_v[(1-\lambda)L^c] + \lambda L^s$, where L^c is the Binary Cross Entropy (BCE) loss, Θ are the trainable parameters, and L^s is a triplet-based objective function that optimizes the similarity between augmentations of the same node, by reducing their cosine distance. It is defined as $L^s = \mathbb{E}_v[\frac{1}{2}(D(t(z_v), sg(\bar{z}_v)) + D(t(\bar{z}_v), sg(z_v))]$, where D is the cosine distance. Finally, NIFTY normalizes the encoder weight matrix at each layer k to impose an upper bound to the change of the original node embeddings.

As for the second approach to enforce fairness on graphs, i.e. to modify the input graphs, EDITS [16] is a model-agnostic framework that ensures fairness with attribute and structural debiasing. Other approaches focus on debiasing the adjacency matrix only, using concepts from optimal transport [17] or dyadic fairness [4]. Among these approaches, in this paper we exploit concepts from Fairdrop, that we explain in detail. Fairdrop [3] is a biased edge dropout algorithm to counter-act homophily (the principle that similar users interact at higher rate than dissimilar ones) of the sensitive attribute to improve fairness in graph embeddings. For every epoch of training, authors propose to: *(i)* compute an $n \times n$ mask M encoding the edge homophily w.r.t. a sensitive attribute, i.e. $m_{i,j} = 1$ if $s_i \neq s_j$, 0 otherwise, *(ii)* define $\tilde{M}$, a random perturbation of M, as $\tilde{m}_{i,j} = m_{i,j}$ with probability $\frac{1}{2} + \delta$, or $1 - m_{i,j}$ with probability $\frac{1}{2} - \delta$, and $\delta \in [0, \frac{1}{2}]$, and *(iii)* drop edges from the original adjacency matrix according to the computed perturbation: $A_{fair} = A \circ \tilde{M}$. The *delta* parameter regulates the level of fairness enforced: when $\delta = 0$, it drops random edges with a probability $p = \frac{1}{2}$, while when $\delta = \frac{1}{2}$, it drops **all** the homophilous edges, keeping only the heterophilous ones. Authors show that removing homophilous edges has a positive effect on different fairness metrics.

4 Method

In this paper, we extend NIFTY [2] using a different algorithm to perturb the graph structure, exploiting the research in input-debiasing methods. While in NIFTY the perturbed adjacency matrix have approximately the same homophily level as the original one, it has been shown that using a perturbation that is biased toward removing homophilous edges can be beneficial for the fairness of the learned model [3]. We use a variation of FairDrop to generate the perturbation. In fact, FairDrop does not have a way to control the amount of perturbation that is inserted, i.e., it is defined to consider all the edges. We implement a variation that used another hyper-parameter, the drop rate, that

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

specifies the ratio of edges that should be perturbed. More formally, given the augmented adjacency matrix $\tilde{A}$ constructed by NIFTY for a certain drop rate, we define $E^{\tilde{A}} = \{(i,j)|(i,j) \in E, \tilde{a}_{i,j} = 1\}$ as the set of edges not dropped and $E^{\neg\tilde{A}} = \{(i,j)|(i,j) \in E, \tilde{a}_{i,j} = 0\}$ as the set of edges dropped. We consider $E^{\neg\tilde{A}}$ and the hyper-parameter δ, and check whether in the set there are at least a ratio of $0.5 + \delta$ edges connecting nodes with the same value of the sensitive attribute (homophilous). If not, we randomly replace some *heterophilous* edges (i.e., edges connecting nodes wth different value for s) from $E^{\neg\tilde{A}}$ and insert them in $E^{\tilde{A}}$, replacing them with *homophilous* edges from $E^{\tilde{A}}$. The number of replaced nodes is given by the difference between the actual ratio of homophilous edges dropped and the given $0.5 + \delta$. Clearly, when in the graph structure the percentage of homophilous edges is already bigger than $0.5 + \delta$ % our extension doesn't activate and reduces itself exactly to NIFTY. On the contrary, the higher the value of δ, the more homophilous links are removed from the adjacency matrix. In this way, the augmented adjacency matrix will contain fewer homophilous links, while maintaining the majority of the edges in the original perturbation from NIFTY: the main difference is given by the selection of edges added and removed.

5 Experimental Evaluation

In our experiments, we considered the version of NIFTY using Graph Convolutional Network (see Section 2) as model backbone. We fixed the network hyper-parameters as per the NIFTY paper, while we explored different values for the drop-rate (Dr) and δ parameters. To study the behavior of our method with different hyper-parameter configurations, we report three values for each of them. Note that the values of δ have to be chosen considering the actual homophily level of the dataset. Only for the German dataset [2], to obtain results that were closer to the ones reported in literature we increased the number of epochs from 1000 to 2500.

Datasets. We tested our solution on four graph datasets. *German credit graph* has 1000 nodes representing clients in a German bank, connected based on the similarity of their credit accounts; clients are to be classified as good or bad credit risks and the sensitive attribute is their gender. (homophily level: 80.92%). *Recidivism graph (Bail)* [2] has 18876 nodes representing defendants who got released on bail, connected based on the similarity of their criminal records and demographics; defendants are to be classified as releasable or not releasable and the sensitive attribute is their ethnicity. (homophily level: 53.61%). *Credit defaulter graph* [2] has 30000 nodes representing individuals, connected based on the similarity of their spending and payment patterns; the task is to predict whether an individual will default on the credit card payment or not and the sensitive attribute is their age. (homophily level : 95.99%). Finally, we consider a dataset where, differently from the others, edges are not generated based on some heuristics. *Pokec* [10] is a dataset representing 66569 users in a social network, connected by friendship relationships; the task is to predict the working field, and the sensitive attribute is the users geographical region. (homophily level: 95.58%).

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Metrics. We use *AUROC* (AUC) to evaluate the predictive performance for the downstream task of node classification. To measure group fairness, we use *Statistical Parity* (SP) and *Equal Opportunity* (EO), where probabilities are estimated on the test set. The third fairness metric we use is *Counterfactual Fairness* (CF), where the score is calculated as the percentage of test nodes for which the predicted label changes when flipping the node's sensitive attribute. We report the fairness metrics as percentages from 0% (never discriminating) to 100% (always discriminating), so the lower the better.

Results and Discussion Tables 1 and 2 present our experimental results. In German (Table 1 left) we see that our method improves on all the considered fairness metrics (SP, EO, and CF) compared to the baseline NIFTY (rows with $\delta = 0$), where $\delta = 0.4$ seems to provide the best results. On Pokec (Table 1 right), we see a similar pattern for SP and CF, while EO improves compared to the baseline for drop rate 0.1 and 0.2. Note however that our method achieves the overall lowest value on EO. Bail (Table 2 left) shows a more complex behavior, where with $Dr = 0.1$ our method improves SP, with $Dr = 0.25$ it improves EO and CF. $Dr = 0.25$ turns out to bit too high for this dataset, where the proposed method improves EO but to a lesser extent compared to 0.25. Finally in Credit (Table 2 right) our method consistently improves all the fairness metrics. In all cases, the decrease in AUC of our method compared to the baseline is always negligible. From the results, it is clear that our method is a powerful extension to NIFTY, being able to significantly improve the fairness of the resulting model for all the metrics and in all the considered datasets. The choice of δ and Dr are however crucial to obtain satisfying results.

6 Conclusions and future works

In this paper, we proposed an extension of NIFTY, a method to compute fair graph node representations. Our proposal is inspired by recently proposed biased graph structure modification methods. We have shown the effectiveness of our approach in four real-world graph datasets. In the future, we will study methods to automatically set optimal values for the hyperparameters of our method and we will study in more depth the changes in representations induced by our proposal, to understand in which cases it results more useful.

		German						Pokec			
Dr	δ	AUC	Δ_{SP}	Δ_{EO}	CF	Dr	δ	AUC	Δ_{SP}	Δ_{EO}	CF
0.01	0	71.05	2.19	3.12	0.30	0.1	0	67.24	0.63	0.70	0.13
		±0.56	±1.52	±1.57	±0.15			±0.43	±0.06	±0.28	±0.03
	.35	71.05	2.15	3.04	0.20		.47	66.42	0.53	0.34	0.10
		±0.57	±1.03	±1.11	±0.28			±0.12	±0.02	±0.02	±0.01
	.4	70.63	1.34	2.35	0.43		.499	66.94	0.57	0.17	0.09
		±0.38	±0.70	±1.06	±0.31			±0.12	±0.05	±0.07	±0.01
0.1	0	68.71	2.87	2.74	0.60	0.2	0	66.52	0.62	0.56	0.13
		±1.06	±1.33	±1.26	±0.18			±0.60	±0.17	±0.36	±0.03
	.35	69.45	2.15	2.36	0.40		.47	66.38	0.48	0.47	0.11
		±1.01	±1.49	±1.67	±0.28			±0.01	±0.02	±0.12	±0.01
	.4	67.17	0.85	0.81	0.50		.499	66.39	0.66	0.74	0.08
		±1.47	±0.57	±0.56	±0.41			±0.04	±0.06	±0.16	±0.01

Table 1: Comparison between NIFTY (rows with $\delta = 0$) and our method on German and Pokec datasets.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

		Bail						Credit			
Dr	δ	AUC	ΔSP	ΔEO	CF	Dr	δ	AUC	ΔSP	ΔEO	CF
0.1	0	81.66 ±0.08	2.26 ±0.19	0.79 ±0.36	1.04 ±0.06	0.01	0	72.13 ±0.01	12.66 ±0.94	10.30 ±0.88	0.78 ±0.75
	.25	82.85 ±0.37	1.40 ±0.18	1.05 ±0.10	1.37 ±0.10		.47	72.09 ±0.01	11.72 ±0.02	9.45 ±0.02	0.08 ±0.01
	.4	83.24 ±0.14	1.67 ±0.18	1.09 ±0.25	1.21 ±0.08		.499	72.09 ±0.02	11.73 ±0.06	9.45 ±0.05	0.10 ±0.02
0.25	0	84.28 ±0.12	2.09 ±0.18	1.11 ±0.59	1.61 ±0.20	0.2	0	72.15 ±0.00	12.84 ±0.02	10.54 ±0.03	0.84 ±0.01
	.25	83.51 ±0.14	2.28 ±0.18	0.61 ±0.31	1.36 ±0.09		.47	72.06 ±0.03	11.73 ±0.05	9.45 ±0.04	0.12 ±0.03
	.4	83.97 ±0.08	2.30 ±0.15	0.44 ±0.12	1.24 ±0.07		.499	72.04 ±0.04	11.75 ±0.03	9.46 ±0.01	0.11 ±0.03

Table 2: Comparison between NIFTY (rows with $\delta = 0$) and our method on Credit and Bail datasets.

References

[1] D. Franco, , N. Navarin, M. Donini, D. Anguita, and L. Oneto. Deep fair models for complex data: Graphs labeling and explainable face recognition. *Neurocomputing*, 470:318–334, 2022.

[2] C. Agarwal, H. Lakkaraju, and M. Zitnik. Towards a unified framework for fair and stable graph representation learning. In *UAI*, 2021.

[3] I. Spinelli, S. Scardapane, A. Hussain, and A. Uncini. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Transactions on Artificial Intelligence*, 2021.

[4] O. Li, Y. Wang, H. Zhao, P. Hong, and H. Liu. On dyadic fairness: Exploring and mitigating bias in graph connections. In *ICLR*, 2021.

[5] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

[6] A. Micheli. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3):498–511, 2009.

[7] F. Scarselli, M. Gori, Ah Chung Ah Chung Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[8] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.

[9] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NeurIPS*, 2016.

[10] S. Dai and S. Wang. Fairgnn: Eliminating the discrimination in graph neural networks with limited sensitive attribute information. *arXiv*, 2020.

[11] A. J. Bose and W. L. Hamilton. Compositional fairness constraints for graph embeddings. *arXiv*, 2019.

[12] A. Buyl and T. De Bie. Debayes: a bayesian method for debiasing network embeddings. *arXiv*, 2020.

[13] B. Kang, J. Lijffijt, and T. De Bie. Conditional network embeddings. In *ICLR*, 2019.

[14] N. Navarin, L. Oneto, and M. Donini. Learning deep fair graph neural networks. In *ESANN*, 2020.

[15] Y. Rong, W. Huang, T. Xu, , and J. Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *ICLR*, 2020.

[16] Y. Dong, N. Liu, B. Jalaian, and J. Li. EDITS: modeling and mitigating data bias for graph neural networks. 2021.

[17] C. Laclau, I. Redko, M. Choudhary, and C. Largeron. All of the fairness for edge prediction with optimal transport. *arXiv*, 2020.