Learning Deep Fair Graph Neural Networks

Nicolò Navarin[1], Luca Oneto[2]*, Michele Donini[3]

[1]University of Padua - Via Trieste 63, 35121, Padova, Italy
[2]University of Genoa - Via Opera Pia 11a, 16145, Genova, Italy
[3]Amazon - 9th Ave Seattle, WA 98101, USA

Abstract. Developing learning methods which do not discriminate subgroups in the population is the central goal of algorithmic fairness. One way to reach this goal is to learn a data representation that is expressive enough to describe the data and fair enough to remove the possibility to discriminate subgroups when a model is learned leveraging on the learned representation. This problem is even more challenging when our data are graphs, which nowadays are ubiquitous and allow to model entities and relationships between them. In this work we measure fairness according to demographic parity, requiring the probability of the possible model decisions to be independent of the sensitive information. We investigate how to impose this constraint in the different layers of a deep graph neural network through the use of two different regularizers. The first one is based on a simple convex relaxation, and the second one inspired by a Wasserstein distance formulation of demographic parity. We present experiments on a real world dataset, showing the effectiveness of our proposal.

1 Introduction

During the last decade, the widespread distribution of automatic systems for decision making is raising concerns that biases in the training data and model inaccuracies can lead to decisions that treat historically discriminated groups unfavourably [1–4]. As a consequence, machine learning models are often required to meet fairness requirements, ensuring the correction and limitation of unfair behaviours and for this reason, in literature, it is possible to find a plethora of different methods to address this issue. These methods can be mainly divided in three families [5]: methods in the first family change a pre-trained model in order to make it more fair (while trying to maintain the classification performance); in the second family we can find methods that enforce fairness directly during the training phase; the third family of methods implements fairness by modifying the data representation and then employs standard machine learning methods. In this work, analogously to [6], we focus on methods which contemporary learn a data representation, with one or more layers, and a desired model from the data [7]. In this context we will exploit the concept of fair representation where the fairness constraint is directly imposed in the representation layers in order to remove the possibility to discriminate subgroups when the final model is learned, leveraging on the learned representation.

This problem is even more challenging when we have to learn a fair representation of graphs. Graphs allow us to model entities and their relationships. Graphs data are naturally ubiquitous, for example in in social networks [8], or constructed from (non)relational data [9]. Learn a data representation expressive enough to describe the entities in the graph and then learn models leveraging on this representation is a fundamental problem in Machine Learning. Many methods exists in literature [10–12] but most of them have several limitations [13]:

*This work was supported by the Amazon AWS Machine Learning Research Award.

some are transductive and do not handle unseen nodes and/or graphs, some are not space-efficient and impractical for large graphs, and some lack support for attributed graphs.

In this work, we will exploit the notion of demographic parity [14] to measure the unfairness, which requires the probability of the possible model decisions to be independent of the sensitive information. Then we will employ, for learning a graph representation, a recently developed Deep Graph Neural Network (DGNN) [13] which addresses many issues of the methods available in literature and has been shown to be very effective in practice. Finally, we will impose this fairness constraint in the different layers of the DGNN, building a Deep Fair Graph Neural Network (DFGNN), though the use of two different regularizers. The first one is based on a convex relaxation of the the demographic parity [6]. The second one is based on a formulation of the the demographic parity based on the Wasserstein distance [15] or, more precisely, the Sinkhorn distance [16], which offers an efficient differentiable alternative to the heavy cost of evaluating the Wasserstein distance directly.

Result on the Pokec[1]dataset [17], the most popular on-line social network in Slovakia, will support our proposal.

2 Deep Fair Graph Neural Network

Let us consider a binary classification problem where a training graph $\mathcal{G}^{\mathrm{Tr}} = (\mathcal{V}, \mathcal{E}, X, \boldsymbol{s}, \boldsymbol{y})$ is given, where $\mathcal{V} = \{v_1, \ldots, v_{d_d}\}$ is the set of d_d nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, $X \in \mathbb{R}^{d_d \times d_x}$ is the matrix of non-sensitive attributes ($\boldsymbol{x}_i$, the i-th row of X, is the vector of attributes associated to the vertex v_i), $s_i \in \{1, 2\}$ is the sensitive attribute[2] associated to node v_i, and $y_i \in \{\pm 1\}$ is label associated to node v_i. Let us define the neighborhood of a vertex v_i as $\mathcal{N}(v_i) = \{v_j | (v_i, v_j) \in E\}$. The training set is composed by all the nodes in the training graph. The goal is to learn a model $\mathsf{h}(\mathsf{Z})$, where, since in some jurisdictions the functional form of h cannot depend, implicitly or explicitly [18], on s, we indicate Z its input composed by v, $\mathcal{E}$, $\boldsymbol{x}$, and possibly s if legally possible. We consider the challenging inductive setting, where two different graphs are taken, one for training and one for testing. In other words, the test is the set of nodes from a second graph $\mathcal{G}^{\mathrm{Te}}$. A dataset, $\mathcal{D}^{\mathrm{Tr}} = \{(\mathsf{Z}_i, s_i, y_i) | i \in \{1, \cdots, d_d\}\}$, is generated from $\mathcal{G}^{\mathrm{Tr}}$ and, analogously, $\mathcal{D}^{\mathrm{Te}}$ is generated from $\mathcal{G}^{\mathrm{Te}}$.

The model h is a composition of models $\mathsf{m}(\mathsf{r}(\mathsf{Z}))$, where $\mathsf{m} : \mathbb{R}^{d_r} \to \mathcal{Y}$ is a linear function and $\mathsf{r}(\mathsf{Z}) \in \mathbb{R}^{d_r}$. r is a function mapping the node, its attributes, and its topological information in the graph into a vector, that we refer to as the representation. Note that r can be a composition of functions too $\mathsf{r} : \mathsf{r}_{d_l} \circ \cdots \circ \mathsf{r}_2 \circ \mathsf{r}_1$, for example, in a deep architectures of d_l layers. In other words, the function r synthesizes the information needed to describe the node in the graph and to learn an accurate model m. Moreover this representation should be, in some sense, fair with respect to the sensitive feature. Specifically, we require that the representation vector satisfies the demographic parity constraint [6, 14]. Namely we require that

$$\mathbb{P}_{\mathsf{Z}}\{\mathsf{r}(\mathsf{Z}) \in \mathbb{C} | s = 1\} = \mathbb{P}_{\mathsf{Z}}\{\mathsf{r}(\mathsf{Z}) \in \mathbb{C} | s = 2\}, \ \forall \mathbb{C} \subseteq \mathbb{R}^{d_r}, \tag{1}$$

[1]We thank Peter Tiňo for its help in handling this dataset.

[2]In this paper, we will consider the case of a single binary valued sensitive attribute.

that is, the two conditional distributions of the representation vector, the one for nodes with $s = 1$ and the one with $s = 2$, should be the same. The accuracy and the fairness of the final models h will be evaluated with the empirical Area Under the Receiver Operating Characteristic curve [19] ($\mathsf{AUROC}_y(\mathsf{h})$), since more informative in case of unbalanced datasets like Pokec, and the empirical Difference of Demographic Parity [6] ($\mathsf{DDP}(\mathsf{h})$)

$$\mathsf{DDP}(\mathsf{h}) = \left| {}^{1}/_{|\mathcal{D}^1|} \sum_{(\mathsf{Z},s,y)\in\mathcal{D}^1} [\mathsf{h}(\mathsf{Z})>0] - {}^{1}/_{|\mathcal{D}^2|} \sum_{(\mathsf{Z},s,y)\in\mathcal{D}^2} [\mathsf{h}(\mathsf{Z})>0] \right|, \quad (2)$$

where the Iverson bracket notation is exploited and $\mathcal{D}^i = \{(\mathsf{Z}, s, y) \in \mathcal{D} | s = i\}$ with $i \in \{1, 2\}$ and $\mathcal{D}$ can be $\mathcal{D}^{\text{Tr}}$ (during the training phase) or $\mathcal{D}^{\text{Te}}$ (in the test phase). Another sanity check on the fairness of $\mathsf{r}(\mathsf{Z})$ is to measure the ability of the same model m to learn s instead of y from $\mathsf{r}(\mathsf{Z})$ itself. Then the accuracy of h in learning s is also measured ($\mathsf{ACC}_s(\mathsf{h})$). Note that imposing the demographic parity in the representation (see Eq. (1)) implies the true DDP of the final model to be zero and this is why imposing a fair representation is more powerful than imposing the fairness of the model since any model built leveraging on a fair representation will be consequently fair.

2.1 Deep Graph Neural Networks

In this paper, we consider the GraphSAGE DGNN model [13], since, contrarily to other architectures in literature (e.g. [20]), it is designed to deal with large graphs (such as social network graphs) and it allows to consider a fixed-size set of neighbors. The representation of a node v at layer k is defined as:

$$\mathsf{r}_{k,v} = \mathsf{ReLU}\left(W_k \cdot \mathsf{mean}\left(\{\mathsf{r}_{k-1,v}\} \cup \{\mathsf{r}_{k-1,u}, \forall u \in \mathsf{sample}(\mathcal{N}(v), n_s)\}\right)\right), \quad (3)$$

where W_k is the matrix of parameters for the k-th layer, ReLU is the rectified linear activation function, mean is the function returning the mean vector over a set of vectors[3], and sample is a function randomly sampling a subset of n_s elements in the set of neighbors $\mathcal{N}(v)$. We then stack multiple (d_l) layers like the one of Eq. (3). For more details about the network, we refer the reader to the original paper [13]. The DGNN has been trained using the Mini Batch Stochastic Gradient Descend, minimizing the empirical Binary Cross-Entropy ($\mathsf{BCE}(\mathsf{h})$).

2.2 Imposing the Demographic Parity

In this section we will propose different approaches for imposing the constraint of Eq. (1) into the DGNN described in Section 2.1. In particular, we propose to add a constraint, though the Tikhonov philosophy [7], as regularizer $\mathsf{F}(\mathsf{h})$ to add in the cost function to be minimzed for training the DGNN, together with the $\mathsf{BCE}(\mathsf{h})$, as follows

$$\mathsf{h}^* = \arg\min_{\mathsf{h}} (1 - \lambda)\mathsf{BCE}(\mathsf{h}) + \lambda\mathsf{F}(\mathsf{h}), \quad (4)$$

where $\lambda \in [0, 1]$ trades off accuracy and fairness as we will also see in Section 3.

The regularizers can act on the representation layers in two different ways. On way is to impose the constraint just on the last layer of the representation,

[3] In the original work [13] mean can be substituted with any aggregation operator.

namely $\mathsf{F}(\mathsf{h}) = \mathsf{F}(\mathsf{r})$. The other way is to impose the constraint on each layer of the representation, namely $\mathsf{F}(\mathsf{h}) = \mathsf{F}(\mathsf{r}_1, \cdots, \mathsf{r}_{d_l})$.

We propose two different regularizers. The first one is based on the work of [6], where a convex approximation and relaxation of the constraint of Eq. (1) is proposed. Analogously to [6], here we propose the following regularizers

$$\mathsf{F}(\mathsf{r}) = {}^1\!/_{d_r} \sum_{i=1}^{d_r} \left| {}^1\!/_{|\mathcal{D}^1|} \sum_{(\mathsf{Z},s,y)\in\mathcal{D}^1} \mathsf{r}_i(\mathsf{Z}) - {}^1\!/_{|\mathcal{D}^2|} \sum_{(\mathsf{Z},s,y)\in\mathcal{D}^2} \mathsf{r}_i(\mathsf{Z}) \right|, \quad (5)$$

which means that the average output, conditioned to the sensitive features, of each element of the representation vector should be the same independently from the sensitive features.

The second approach that we propose, requires the definition of the following probability density functions[4]

$$\mathsf{p}_1(\boldsymbol{t}) = \mathbb{P}_{\mathsf{Z}}\{\mathsf{r}(\mathsf{Z}) = \boldsymbol{t} | s = 1\},\ \mathsf{p}_2(\boldsymbol{t}) = \mathbb{P}_{\mathsf{Z}}\{\mathsf{r}(\mathsf{Z}) = \boldsymbol{t} | s = 2\},\ \boldsymbol{t} \in \mathbb{R}^{d_r}. \quad (6)$$

Then we impose that the two empirical counterparts of these distributions, respectively $\hat{\mathsf{p}}_1(\boldsymbol{t})$ and $\hat{\mathsf{p}}_2(\boldsymbol{t})$, should be close to each other with respect to the Wasserstein distance [15]

$$\mathsf{F}(\mathsf{r}) = \mathsf{W}(\hat{\mathsf{p}}_1, \hat{\mathsf{p}}_2). \quad (7)$$

Note that if $\mathsf{W}(\mathsf{p}_1, \mathsf{p}_2) = 0$ we also have that the constraint of Eq. (1) is satisfied. Note also that Eq. (7) is hard to impose since it is computationally expensive. For this reason we will use the Sinkhorn distance [16], which offers an efficient differentiable alternative to the heavy cost of evaluating the Wasserstein distance directly.

Finally, for $\mathsf{F}(\mathsf{r}_1, \cdots, \mathsf{r}_{d_l})$, namely for imposing the fairness constraint to all the layers of the network, we simply apply the same regularizers of Eqns. (5) and (7) to each layer composing the representation.

3 Results and Discussion

In this section, we present the results of our experiments on the Pokec dataset [21]. Pokec was the most popular on-line social network in Slovakia. The released dataset contains anonymized data of the whole network. Profile data includes gender, age, marital status, and other information. We selected the sub-network composed by the users with a specified gender, age, and marital status (we discarded the other users). We choose the marital status as the target feature – or node label – and the gender as the sensitive attribute. We then randomly split the users in 2 halves assigning them to $\mathcal{D}^{\text{Tr}}$ (training set) and $\mathcal{D}^{\text{Te}}$ (test set). As stated in Section 2, we deal with this problem in the inductive setting, in contrast with the semi-supervised learning one [20], in which it is assumed to know the whole network structure beforehand. We thus train our model on the training network, that comprehends nodes and relationships only about the users in $\mathcal{D}^{\text{Tr}}$. We then predict the target feature for the nodes in $\mathcal{D}^{\text{Te}}$. We consider a graph neural network with two GraphSAGE layers with uniform sampling of $n_s = 25$ neighbouring nodes, and with 128 hidden neurons. The output fully connected layer has size 2 with a softmax activation function.

The results of our experiments are reported in Fig. 1, 2, and 3, and described in their captions and legends. From these results it is possible to observe that:

[4] Note that this definition is formally correct only if $\mathsf{r}(\mathsf{Z})$ assumes values in a finite set.

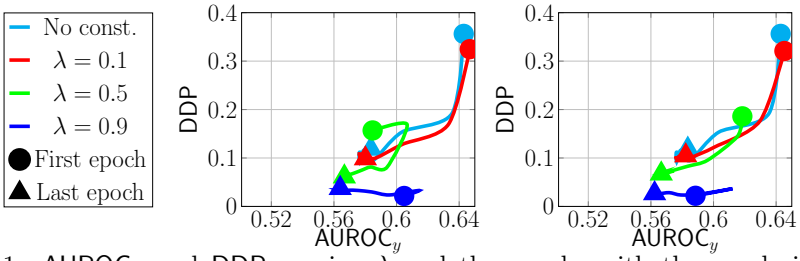

Fig. 1: AUROC$_y$ and DDP varying λ and the epochs with the regularizer of Eq. (5) which acts (Left) on the last layer or (Right) on each layer of the representation. The sensitive feature is in the functional form of the model.

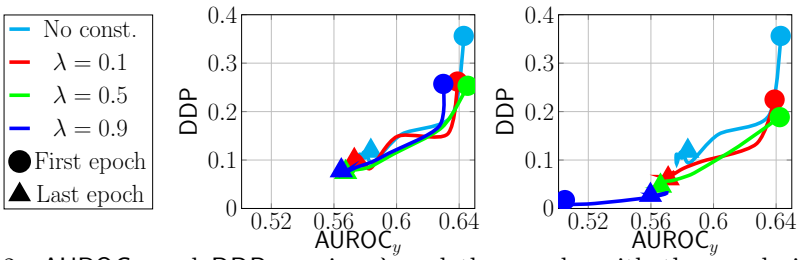

Fig. 2: AUROC$_y$ and DDP varying λ and the epochs with the regularizer of Eq. (7) which acts (Left) on the last layer or (Right) on each layer of the representation. The sensitive feature is in the functional form of the model.

- adding the constraint, both Eq. (5) or Eq. (7), remarkably improves the fairness (DDP) of the solution without compromising its accuracy (AUROC$_y$);
- applying the constraint to all the representation layers is more effective than applying it just to the last layer (without compromising its accuracy);
- the new constraint Eq. (7) is more effective than the known one of Eq. (5);
- not exploiting the sensitive feature in the functional form generates models that are fairer and less accurate. As expected – also in this scenario – the model generated using $\lambda = 0.9$ is the most fair and it keeps a good accuracy;
- measuring fairness with the ACC$_s$, instead of the DDP, shows that the sensitive feature reconstruction error is larger when the constraint is active;
- in the first epochs fairness decreases without compromising the accuracy; the user can decide to stop the training as soon as the model respects the desired trade-off between accuracy and fairness.

Even if the results are quite promising, this work is just a fist step toward the solution of the problem of learning fair models for graphs since the proposed approach should be tested on different datasets and the methods should be supported by deeper theoretical insights (e.g. consistency in terms of accuracy and fairness).

References

[1] B. Cowgill. Bias and productivity in humans and algorithms: theory and evidence from resume screening. In *Columbia Business School, Columbia University*, 2018.
[2] S. Barocas and A. D. Selbst. Big data's disparate impact. *California Law Review*, 104:671, 2016.
[3] I. Deborah Raji and J. Buolamwini. Actionable auditing: investigating the impact of

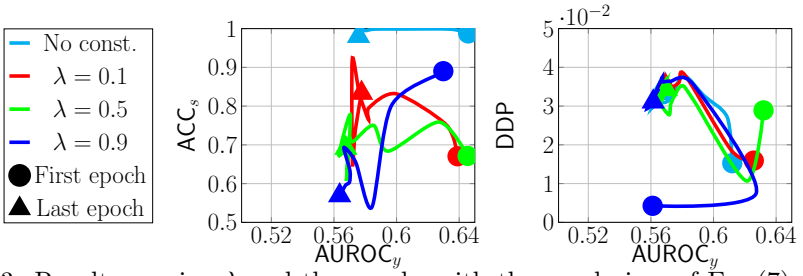

Fig. 3: Results varying λ and the epochs with the regularizer of Eq. (7) which acts on the last layer of the representation. (Left) AUROC_y and ACC_s when the sensitive feature is in the functional form of the model. (Right) AUROC_y and DDP when the sensitive feature is not in the functional form of the model.

publicly naming biased performance results of commercial ai products. In *AAAI/ACM Conference on AI Ethics and Society*, 2019.

[4] J. Buolamwini and T. Gebru. Gender shades: intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, 2018.

[5] M. Donini, L. Oneto, S. Ben-David, J. Shawe-Taylor, and M. Pontil. Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[6] L. Oneto, M. Donini, A. Maurer, and M. Pontil. Learning fair and transferable representations. *arXiv preprint arXiv:1906.10673*, 2019.

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[8] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in facebook. In *ACM workshop on Online social networks*, 2009.

[9] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[10] R. A. Rossi, R. Zhou, and N. K. Ahmed. Deep feature learning for graphs. *arXiv preprint arXiv:1704.08829*, 2017.

[11] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: online learning of social representations. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[12] A. Grover and J. Leskovec. node2vec: scalable feature learning for networks. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016.

[13] W. Hamilton, Zhitao Ying, and J. Leskovec. Inductive representation learning in large attributed graphs. In *Neural Information Processing Systems*, 2017.

[14] S. Verma and J. Rubin. Fairness definitions explained. In *IEEE/ACM International Workshop on Software Fairness*, 2018.

[15] C. Villani. *Optimal transport: old and new*. Springer Science & Business Media, 2008.

[16] M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Neural Information Processing Systems*, 2013.

[17] L. Takac and M. Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, 2012.

[18] C. Dwork, N. Immorlica, A. T. Kalai, and M. D. M. Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, 2018.

[19] C. D. Brown and H. T. Davis. Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 80(1):24–38, 2006.

[20] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[21] L. Takac and M. Zabovsky. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, 2012.