The KL-Divergence between a Graph Model and its Fair I-Projection as a Fairness Regularizer

Maarten Buyl and Tijl De Bie
IDLab, Ghent University
{maarten.buyl, tijl.debie}@ugent.be

Abstract

Learning and reasoning over graphs is increasingly done by means of probabilistic models, e.g. exponential random graph models, graph embedding models, and graph neural networks. When graphs are modeling relations between people, however, they will inevitably reflect biases, prejudices, and other forms of inequity and inequality. An important challenge is thus to design accurate graph modeling approaches while guaranteeing fairness according to the specific notion of fairness that the problem requires. Yet, past work on the topic remains scarce, is limited to debiasing specific graph modeling methods, and often aims to ensure fairness in an indirect manner.

We propose a generic approach applicable to most probabilistic graph modeling approaches. Specifically, we first define the class of fair graph models corresponding to a chosen set of fairness criteria. Given this, we propose a fairness regularizer defined as the KL-divergence between the graph model and its I-projection onto the set of fair models. We demonstrate that using this fairness regularizer in combination with existing graph modeling approaches efficiently trades-off fairness with accuracy, whereas the state-of-the-art models can only make this trade-off for the fairness criterion that they were specifically designed for.

1. Introduction

Graphs are flexible data structures, naturally suited for representing relations between people (e.g. in social networks) or between people and objects (e.g. in recommender systems). Here, links between nodes may represent any kind of relation, such as interest or similarity. It is common in real-world relational data that the corresponding graphs are often imperfect or only partially observed. For example, it may contain spurious or missing edges, or some node pairs may be explicitly marked as having unknown status. In such cases, it is often useful to correct or predict the link status between any given pair of nodes. This task is known as *link prediction*: predicting the link status between any pair of nodes, given the known part of the graph and possibly any node or edge features (Liben-Nowell & Kleinberg, 2007).

Methods for link prediction are typically based on machine learning. A first class of methods constructs a set of features for each node-pair, such as their number of common neighbours, the Jaccard similarity between their neighbourhoods, and more (Martínez et al., 2016). Other methods are based on probabilistic models, with exponential random graph models as a notable class originating mostly from the statistics and physics communities (Robins et al., 2007). More recently, the machine learning community has proposed graph embedding methods (Hamilton et al., 2017), which represent each node as a point in a vector space, from which a probabilistic model for the graph's edges can be derived (among other possible uses). Related to this, graph neural network models (Wu et al., 2020) have been proposed which equally can be used to probabilistically model the presence or absence of edges in a graph (Zhang & Chen, 2018).

The use of such models can have genuine impact on the lives of the individuals concerned. For example, a graph of data on job seekers and job vacancies can be used to determine which career opportunities an individual will be recommended. If it is a social network, it may determine which friendships are being recommended. The existence of particular undesirable biases in such networks (e.g. people with certain demographics being recommended only certain types of jobs, or people with a certain social position only being recommended friendships with people of similar status) may result in biased link predictions that perpetuate inequity in society. Yet, graph models used for link prediction typically exploit properties of graphs that are a direct or indirect result of those existing biases. For example, many will exploit the presence of *homophily*: the tendency of people to associate with similar individuals (McPherson et al., 2001). However, homophily leads to segregation, which often adversely affects minority groups (Hofstra et al., 2017; Karimi et al., 2018).

The mitigation of bias in machine learning algorithms has been studied quite extensively for classification and regression models in the fairness literature, both in formalizing a range of fairness measures (Dwork et al., 2012; Hardt et al., 2016) and in developing methods that ensure fair classification and regression (Mehrabi et al., 2019). However, despite the existence of biases, such as homophily, that are specific to relational datasets, fairness has so far received limited attention in the graph modeling and link prediction literature. Current approaches focus on resolving bias issues for specific algorithms (Buyl & De Bie, 2020; Bose & Hamilton, 2019), or use adversarial learning to improve a specific notion of fairness (Masrour et al., 2020; Bose & Hamilton, 2019).

There is thus a need for further study into fair link prediction, both in applicability to generic network models, and to more generic notions of fairness. To that end, we present the following contributions in this paper:

- We express the class of all *fair probabilistic network models* in Sec. 3 and then use this characterization to compute the *fair I-projection*, the fair distribution that has the smallest KL-divergence with a reference distribution.
- In Sec. 4, we propose the KL-divergence of a distribution with its fair I-projection as a *fairness regularizer*, to be added to the cross-entropy cost function of any probabilistic graph model.
- Finally, the results of our experiments in Section 5 demonstrate that our proposed fairness regularizer *can be applied to a wide diversity of probabilistic graph models* such that the desired fairness score is improved. In terms of that fairness criterion, our fairness modification outperforms *DeBayes* and *Compositional Fairness Constraints* on the models they were specifically designed for.

2. Related Work

Fairness-aware machine learning is traditionally divided into three types (Mehrabi et al., 2019): *pre-processing* methods that involve transforming the dataset such that bias is removed (Calmon et al., 2017), *in-processing* methods that try to modify the algorithm itself and *post-processing* methods that transform the predictions of the model (Hardt et al., 2016). Our method belongs to the in-processing category, because we directly modify the objective function with the aim of improving fairness. Here, one approach is to enforce constraints that keep the algorithm fair throughout the learning process (Woodworth et al., 2017).

Alternatively, the fairness-constrained optimisation problem can be solved using the method of Lagrange multipliers (Agarwal et al., 2018; Jiang & Nachum, 2020; Cotter et al., 2019; Wei et al., 2020). They are related to the problem of finding the fair I-projection (Csiszár & Matus, 2003): the distribution from the set of fair distributions with the smallest KL-divergence to some reference distribution, e.g. an already trained (biased) model (Alghamdi et al., 2020). While we also compute the I-projection of the model onto the class of fair link predictors, we do not use it to transform the model directly. Instead, we consider the distance to that I-projection as a regularization term.

The work on applying fairness methods to the task of link prediction is sparse. Methods *DeBayes* (Buyl & De Bie, 2020) and *Fairwalk* (Rahman et al., 2019) both adapt specific graph embedding models to make them more fair. Other approaches, e.g. *FLIP* (Masrour et al., 2020) and *Compositional Fairness Constraints* (Bose & Hamilton, 2019), rely on adversarial learning to remove sensitive information from node representations.

3. Fair Information Projection

Given a characterization of the set of fair graph models, according to a chosen fairness criterion, a simple approach for finding a fair graph model would be to first fit the chosen graph model to the empirical graph, and to subsequently replace it with its I-projection onto this set.

After discussing some notation in Sec. 3.1, we will investigate this set of fair graph models in Sec. 3.2 and argue that for common fairness metrics, the set of fair graph models is a linear set. The I-projection on such sets, discussed in Sec. 3.3, is well-studied and easy to compute in practice.

3.1. Notation

We denote a random unweighted and undirected graph without self-loops as $G = (V, E)$, with $V = \{1, 2, \ldots, n\}$ the set of n vertices and $E \subseteq \binom{V}{2}$ the set of edges. It is often convenient to represent the set of edges also by a symmetric adjacency matrix with zero diagonal $\mathbf{A} \in \{0, 1\}^{n \times n}$ with element a_{ij} at row i and column j equal to 1 if $\{i, j\} \in E$ and 0 otherwise. An empirical graph over the same set of vertices will be denoted as $\hat{G} = (V, \hat{E})$ with adjacency matrix $\hat{\mathbf{A}}$ and adjacency indicator variables $\hat{a}_{ij}$. In some applications, $\hat{a}_{ij}$ may be unobserved and thus unknown for some $\{i, j\}$.

A probabilistic graph model p for a given vertex set V is a probability distribution over the set of possible edge sets E, or equivalently over the set of adjacency matrices $\mathbf{A}$, with $p(\mathbf{A})$ denoting the probability of the graph with adjacency matrix $\mathbf{A}$. Probabilistic graph models are used for various purposes, but one important purpose is link prediction: the prediction of the existence of an edge (or not) connecting any given pair of nodes i and j. This is particularly important when some elements from $\hat{\mathbf{A}}$ are unknown. But it is also useful when the empirical adjacency matrix is assumed to be noisy, in which case link prediction is used to reduce the noise. Link prediction can be trivially done by making use of the marginal probability distribution p_{ij}, defined as $p_{ij}(x) = \sum_{\mathbf{A}:a_{ij}=x} p(\mathbf{A})$.

Note that many practically useful probabilistic graph models are dyadic independence models: they can be written as the product of the marginal distributions: $p(\mathbf{A}) = \prod_{i<j} p_{ij}(a_{ij})$. This is true for the models evaluated in our empirical results section, but the approach proposed in this paper is conceptually applicable also where this is not the case (e.g. for more complex random graph models), albeit at the cost of greater mathematical and computational complexity.

Finally, we assume vertices belong to one of a set of *sensitive groups*, defined by categorical attributes with respect to which discrimination is undesirable or forbidden. These sensitive groups are denoted as V_s with $s \in S$ for some finite set S. The sets V_s with $s \in S$ form a partition of V. For notational convenience, we also introduce the notation $U_{st} \triangleq \{\{i,j\} | i \in V_s, j \in V_t, i \neq j\}$, the set of possible unordered pairs of distinct vertex pairs between V_s and V_t. Thus, $|U_{ss}| = \binom{|V_s|}{2}$ and $|U_{st}| = |V_s| \times |V_t|$ for $s \neq t$. Similarly, we write $U \triangleq \binom{V}{2}$ for the set of all (unordered) vertex pairs.

3.2. Fairness Constraints

Here we take inspiration from prior work (Buyl & De Bie, 2020) on translating two classification fairness criteria to the graph setting: *demographic parity* in Sec. 3.2.1 and *equalised opportunity* in Sec. 3.2.2. For each, we then formalize the set of graph distributions that are fair with respect to that criterion, before generalizing these formalizations in Sec. 3.2.3.

3.2.1. Demographic Parity (DP)

A classifier could be thought of as non-discriminatory when its expected score of an individual is the same regardless of which sensitive group they belong to. This traditional criterion of fairness is referred to as *demographic* or *statistical parity* (DP) (Dwork et al., 2012).

We generalize this to the graph setting by requiring that the expected proportion of vertex pairs belonging to any two sensitive groups V_s and V_t that are connected, is constant over all pairs of sensitive groups. More formally, the probabilistic graph model p satisfies the DP fairness criterion iff:

$$\forall s,t \in S : \mathop{\mathbb{E}}_{\mathbf{A}\sim p}\left[\frac{1}{|U_{st}|}\sum_{\{i,j\}\in U_{st}} a_{ij}\right] = d,$$

for some $d \in \mathbb{R}$. Thanks to linearity of the expectation operator, and with p_{ij} the marginal distribution for the edge indicator variable a_{ij}, this can be simplified as follows:

$$\forall s,t \in S : \sum_{\{i,j\}\in U_{st}} \mathop{\mathbb{E}}_{a_{ij}\sim p_{ij}}[a_{ij}] = d|U_{st}|.$$

We thus define the set $\mathbb{P}_{\text{DP}}$ of distributions satisfying these constraints as fair with respect to DP.

In fair link prediction, the DP fairness criterion is notable for diminishing the effect of homophily, since it encourages inter-group interaction to have the same expected score as intra-group interactions, thereby reducing segregation based on the nodes' sensitive traits.

3.2.2. Equalised Opportunity (EO)

A drawback of the DP fairness criterion is that it disregards the possibility that there are non-discriminatory reasons for some sensitive groups to be ranked higher (Hardt et al., 2016). An example in the social graph context is that one sensitive group may generally have more social interactions with others, regardless of their sensitive group (Buyl & De Bie, 2020). Depending on the application, it may then be deemed fair to score inter-group edges to this more social group higher than intra-group edges between nodes in other groups.

A fairness criterion that takes this into account is *equalised opportunity* (EO) (Hardt et al., 2016). EO requires that the true positive rate, and consequently also the false negative rate, is equal across groups. In other words, and applied to the graph context: when averaging the probability of edge-connected vertex-pairs between two sensitive groups V_s and V_t, the result should always be the same irrespective of s and t. More formally:

$$\forall s,t \in S : \mathop{\mathbb{E}}_{\mathbf{A}\sim p}\left[\frac{1}{|\hat{E}\cap U_{st}|}\sum_{\{i,j\}\in \hat{E}\cap U_{st}} a_{ij}\right] = d,$$

for some $d \in \mathbb{R}$. Thanks to linearity of the expectation operator, and with p_{ij} the marginal distribution for the edge indicator variable a_{ij}, this can be simplified as follows:

$$\forall s,t \in S : \sum_{\{i,j\}\in \hat{E}\cap U_{st}} \mathop{\mathbb{E}}_{a_{ij}\sim p_{ij}}[a_{ij}] = d|\hat{E}\cap U_{st}|.$$

We thus define the set $\mathbb{P}_{\text{EO}}$ of distributions satisfying these constraints as fair with respect to EO.

3.2.3. General Sets of Fair Graph Distributions

Both the DP and EO criteria are thus formalized as a constraint that is linear in the probabilisty distribution p. Using

$\mathbf{1}$ to denote the indicator function, the DP and EO constraints on p can both be formalized in the following form:

$$F_c(p) \triangleq \sum_{\{i,j\}\in U} \mathop{\mathbb{E}}_{a_{ij}\sim p_{ij}} [f_c(\{i,j\}, a_{ij})] = d_c, \quad (1)$$

where for DP the functions $f_c : U \times \{0, 1\} \to \mathbb{R}$ and corresponding constants d_c are given by:

$$f_{st}(\{i,j\}, x) = x\mathbf{1}(\{i,j\} \in U_{st}),$$
$$d_{st} = d|U_{st}|,$$

for all $s, t \in S$ and for some $d \in \mathbb{R}$. Similarly, for EO:

$$f_{st}(\{i,j\}, x) = x\mathbf{1}(\{i,j\} \in \hat{E} \cap U_{st}),$$
$$d_{st} = d|\hat{E} \cap U_{st}|.$$

As a matter of fact, many other fairness criteria, such as *accuracy equality* or *churn equality* can formalized in this manner, with different choices for f_c and d_c (Cotter et al., 2019; Alghamdi et al., 2020; Agarwal et al., 2018).

Thus, although our implementation and experiments are focused on DP and EO only, we develop the theory in this paper for the general formulation of a set of fair probabilistic graph models as:[1]

$$\mathbb{P}_{\mathcal{F}} := \{p \in \mathbb{P} \mid \forall c \in \mathcal{C}_{\mathcal{F}} : F_c(p) = d_c\}, \quad (2)$$

with $\mathbb{P}$ the set of all possible distributions over $\mathbf{A}$, and $\mathcal{C}_{\mathcal{F}}$ a countable (and typically finite) set indexing the constraints that enforce fairness criterion $\mathcal{F}$. Importantly, F_c as defined in Eq. (1) is a linear function of p, such that I-projecting any distribution onto $\mathbb{P}_{\mathcal{F}}$ is a mathematically elegant operation. This is the subject of the following.

3.3. Information Projection

We now show how to find, for any possibly unfair distribution h, the fair distribution $p \in \mathbb{P}_{\mathcal{F}}$ that is as close to h as possible. When that closeness is computed in terms of the KL-divergence, then the desired distribution, denoted by $h_{\mathcal{F}}$, is known as the *I-projection* (Csiszár, 1975; Csiszár & Matus, 2003):

$$h_{\mathcal{F}} = \mathop{\arg\min}_{p\in\mathbb{P}_{\mathcal{F}}} D_{KL}(p \,||\, h),$$

where it is assumed that $\mathbb{P}_{\mathcal{F}} \neq \emptyset$ and $D_{KL}(p \,||\, h) < \infty$. Since $\mathbb{P}_{\mathcal{F}}$ is linear and thus convex, the I-projection $h_{\mathcal{F}}$ is unique (Csiszár & Matus, 2003).

Finding the I-projection of model h under linear constraints $\mathcal{C}_{\mathcal{F}}$ is a convex optimization problem. Although it is straightforward to generalize this, let us assume that h is a dyadic independence model. This is justified as many contemporary probabilistic graph models (including graph embedding methods and graph neural networks) are dyadic independence models, and because it simplifies notation. Then, the I-projection of h is the product distribution of the marginal distributions for the vertex pairs $\{i, j\}$, given by (Cover, 1999):

$$h_{\mathcal{F},ij}(x) = \frac{h_{ij}(x)}{Z_{\mathcal{F},ij}(\lambda)} \exp\left(\sum_{c\in\mathcal{C}_{\mathcal{F}}} \lambda_c f_c(\{i,j\}, x)\right),$$

with

$$Z_{\mathcal{F},ij}(\lambda) = \sum_{x\in\{0,1\}} h_{ij}(x) \exp\left(\sum_{c\in\mathcal{C}_{\mathcal{F}}} \lambda_c f_c(\{i,j\}, x)\right).$$

the *log-partition* function and with λ denoting the vector of λ_c values. Let $Z_{\mathcal{F}}(\lambda) = \prod_{\{i,j\}\in U} Z_{\mathcal{F},ij}(\lambda)$. The values of the λ_c are found by maximizing:

$$L_h(\lambda) = -\log Z_{\mathcal{F}}(\lambda) + \sum_{c\in\mathcal{C}_{\mathcal{F}}} \lambda_c d_c. \quad (3)$$

This function $L_h(\lambda)$ is the Lagrange dual of the KL-divergence minimization problem with reference model h, and λ is the set of Lagrange multipliers corresponding to the fairness constraints.

4. The KL-divergence to the I-projection as a Fairness Regularizer

We argue that the KL-divergence $D_{KL}(h_{\mathcal{F}} \,||\, h)$ between a probabilistic graph model h and its fair I-projection $h_{\mathcal{F}}$ is an adequate measure of the unfairness of h.

Indeed, suppose that $h_{\mathcal{F}}$ represents an idealized version of reality that is free from undue bias (i.e. fair). Specifically, it is the idealized version of reality that is closest to the model h, which, in turn, can be seen as the unfairly biased version of the reality $h_{\mathcal{F}}$. For example, it may be the result of discrimination and cultural social biases in historical data. Then the KL-divergence $D_{KL}(h_{\mathcal{F}} \,||\, h)$ quantifies the amount of information lost when using the biased model h instead of the idealized model $h_{\mathcal{F}}$ (Burnham & Anderson, 1998). In other words, it is the information lost due to any unfairness in the model h, and thus, informally speaking, the amount of 'unfair information' in h.

Moreover, the KL-divergence, in being a measure of information, is commensurate with commonly used loss terms in machine learning, in particular with the cross-entropy between the empirical distribution and the learned model, which is equivalent to the KL-divergence between those two up to a constant. This is the topic of the next subsection.

[1]In our proposed framework, we require these constraints to be satisfied exactly in order for p to be fair. However, prior work has also allowed for a percentage-wise deviation (Zafar et al., 2017).

Algorithm 1 Optimising $\mathcal{L}$ with respect to link predictor h, in the case where DP is the fairness criterion.

Input: possible distinct vertex pairs U, empirical adjacency matrix $\hat{\mathbf{A}}$, and fairness strength parameter γ.
initialize model h and I-projection parameters λ
for $t = 1$ **to** T **do**
  $\mathcal{L}_{\mathcal{A}} \leftarrow -\log h\left(\hat{\mathbf{A}}\right)$
  $d \leftarrow \frac{1}{|U|}\,\mathbb{E}_{\mathbf{A}\sim h}\left[\mathbf{A}\right]$
  $\mathcal{L}_{\mathcal{F}} \leftarrow \max_{\lambda}\left[-\log Z_{h_{\mathcal{F}}}(\lambda) + \sum_{s,t\in S}\lambda_{st} d|U_{st}|\right]$
  $\mathcal{L} \leftarrow \mathcal{L}_{\mathcal{A}} + \gamma\mathcal{L}_{\mathcal{F}}$
  $\mathrm{UPDATE}(h, \nabla_h \mathcal{L})$
end for

4.1. I-Projection Regularization

Let $\hat{p}$ represent the empirical distribution, i.e. $\hat{p}(\mathbf{A} = \hat{\mathbf{A}}) = 1$ and $\hat{p}(\mathbf{A} \neq \hat{\mathbf{A}}) = 0$. The common machine learning objective is then to minimize the KL-divergence $D_{KL}(\hat{p} \,||\, h)$, denoted by $\mathcal{L}_{\mathcal{A}}$, which is equivalent to maximizing the log-likelihood of h under $\hat{p}$, or equivalently the cross-entropy. We propose to add the KL-divergence $D_{KL}(h_{\mathcal{F}} \,||\, h)$ as an extra loss term $\mathcal{L}_{\mathcal{F}}$. The overall objective function $\mathcal{L}$ to find h is thus:

$$\begin{aligned}\mathcal{L} &= \min_h \left[\mathcal{L}_{\mathcal{A}} + \gamma\mathcal{L}_{\mathcal{F}}\right] \\ &= \min_h \left[D_{KL}(\hat{p} \,||\, h) + \gamma D_{KL}(h_{\mathcal{F}} \,||\, h)\right]\end{aligned}$$

with γ a hyperparameter that controls the strength of the loss term. Recall that, for a parameter λ that satisfies the fairness constraints, $D_{KL}(h_{\mathcal{F}} \,||\, h)$ is equivalent to the loss function in Eq. (3):

$$\begin{aligned}\mathcal{L} &= \min_h \left[D_{KL}(\hat{p} \,||\, h) + \gamma \min_p D_{KL}(p \,||\, h)\right] \\ &= \min_h \left[D_{KL}(\hat{p} \,||\, h) + \gamma \max_\lambda L_h(\lambda)\right].\end{aligned}$$

For the difference between this objective and prior work, see Sec. 2.

4.2. Practical Considerations

Out of several ways to optimise $\mathcal{L}$, we opted to fully optimize λ for every parameter update of h. The optimisation of λ can be efficient for several reasons: it is convex, the optimal λ for the previously considered h can be used as the starting guess for new h updates, and the parameter components λ_c are few in number (for DP and EO, there are only $\mathcal{C}_{\mathcal{F}} = |S|^2$). The memory requirements are therefore already limited, while the computational cost can be further reduced by tolerating a less strong convergence for λ or by subsampling h when optimizing λ.

Furthermore, we did not yet specify the choice of d in the DP and EO constraints. To enforce $p \in \mathbb{P}_{\mathcal{DP}}$, a straightforward option is to set d equal to the mean of p. However, this slightly complicates the computation of the I-projection, as d is then no longer constant with respect to p. Alternatively, setting d equal to the mean of the empirical distribution $\hat{p}$ forces p to adopt the same mean as the empirical one, though there is no specific reason that $h_{\mathcal{F}}$ or consequently h should match the empirical mean. We finally chose to set d equal to the mean of h, such that when optimizing λ, we can treat d as a fixed, constant value.

An application of the fairness regularizer to the DP fairness criterion is summarised in Alg. 1.

5. Experiments

Our experiments were performed on three datasets, described in Sec. 5.1. We applied our proposed fairness regularizer on four simple, yet diverse methods explained in Sec. 5.2. Though the method variants without fairness regularizer are already baselines, we additionally compared our results with state-of-the-art approaches for link prediction based on fair graph embedding in Sec. 5.3. All methods went through the same evaluation pipeline described in Sec. 5.4. The results of which were discussed in Sec. 5.5.

5.1. Datasets

The methods were evaluated on three datasets of attributed relational data, summarised in Tab. 1. They were chosen for their diverse properties and manageable size.

POLBLOGS (Adamic & Glance, 2005)
The POLBLOGS dataset was constructed from blogs discussing United States politics in 2005. In the undirected version, there is an edge between blogs if either of them had a hyperlink to the other. The sensitive attribute is the US political *party* (the *Republican* or *Democratic Party*) that the blog supported, either by their own admission or through manual labelling from the dataset creators. Intra-group links are heavily favoured over inter-group links.

ML100K
Movielens datasets are often used as a benchmark for recommender systems. The data contains users' movie ratings on a five-star scale. An unweighted, bipartite graph is formed by considering the users and movies as nodes and an edge between them if the user rated the movie. While the data contains several types of sensitive attributes, we opted to group the *age* attribute into seven bins, delineated by the ages $[18, 25, 35, 45, 50, 56]$. There are only user-movie edges, so the domain of sensitive value of an edge is only affected by the user's sensitive value. Note that all methods were adapted such that they took the bipartitiness of the graph into account when sampling negative training edges.

Table 1: Properties of the datasets. The dataset names are URLs to hosts of the datasets.

DATASET	#NODES	#EDGES	S	$\|S\|$
POLBLOGS	1,222	16,714	PARTY	2
ML100K	2,625	100,000	AGE	7
FACEBOOK	3,955	85,482	GENDER	2

FACEBOOK (McAuley & Leskovec, 2012)
The FACEBOOK graph consists of user nodes that are linked if they are 'friends'. Each user either has *gender* feature '0', '1' or neither. For the last group of users, of which there are 84, it is unclear whether their gender is unknown or non-binary. Their nodes and edges were removed from the dataset. Only 3 undirected attribute pairs thus remain in the data. In contrast to POLBLOGS, the bias effect is much weaker.

5.2. Algorithms

The proposed fairness regularizer was applied to four relatively simple graph models. A *PyTorch* implementation was sought or implemented for each of them, such that the fairness loss can easily be added.

MAXENT
We will refer to the MAXENT model as the maximum entropy graph model under which the expected degree of each node matches its empirical degree (De Bie, 2011). The solution is a simple exponential random graph model (Robins et al., 2007).

DOT-PRODUCT
Given a set of embeddings, one for every node, taking the DOT-PRODUCT an embedding pair is a straightforward way to perform link prediction (Hamilton et al., 2017). In this simple model, the 'decoder' for edge (i, j) is the dot product operator, while the 'encoder' for node i just looks up its representation in a learned table of embeddings.

CNE
A method that combines both the MAXENT model and the DOT-PRODUCT decoder is the *Conditional Network Embedding* (CNE) model (Kang et al., 2018). Instead of the dot-product, it 'decodes' the distance between nodes (i, j). Moreover, it uses the MAXENT model as a prior distribution over the graph data.

GAE
Finally, the Graph Auto-Encoder (GAE) (Kipf & Welling, 2016) is also a DOT-PRODUCT model, though it uses a Graph Convolutional Network (GCN) as its encoder. As such, it is a simple example of a graph neural network (Wu et al., 2020). In our implementation we used two layers for the GCN and used the identity matrix as the node feature matrix.

5.3. Fair Graph Embedding Baselines

In part, the algorithms from Section 5.2 were chosen such that they allow for easy comparison with two recent methods in the field of fair graph embedding.

CFC
The Compositional Fairness Constraints (CFC) method (Bose & Hamilton, 2019) aims to generate fair embeddings by learning filters that mask the sensitive attribute information. This is done through adversarial learning. When applied to link prediction, it also uses the DOT-PRODUCT decoder. Note that our implementation of the basic DOT-PRODUCT differs from the source code of CFC, causing differences in performance between our DOT-PRODUCT experiments and CFC with a fairness regularisation strength of zero.

DEBAYES
Finally, DEBAYES (Buyl & De Bie, 2020) is an adaptation of CNE where the bias in the data is used as additional prior information when learning the embeddings, such that the embeddings are debiased. By using a prior without this biased information at testing time, the link prediction using these embeddings is expected to at least not be less fair than the standard CNE.

5.4. Evaluation

Every method was run for 10 different random seeds on each dataset. Those 10 seeds each had a different train/test split, where the latter consisted of around 20% of the edges in the data. The test set was extended with the same amount of non-edges. However, it was made sure that the test set did not contain nodes unknown in the train set. Only test set results are reported.

Hyperparameter tuning in order to improve the performance of the considered methods was minimal, as our aim is to show the effect of the fairness regularisation and not the predictive quality of the method themselves. As such, we did no hyperparameter sweep with the aim of improving AUC, and instead only deviated from default parameters when it could allow for an easier comparison between models, e.g. the dimensionality of DOT-PRODUCT and CFC embeddings. We only report results of our proposed method with a fairness regularisation strength of $\gamma = 100$, because this parameter almost always caused a significant effect on the fairness measures while not diminishing predictive power too strongly. For DEBAYES the default values were used, while for CFC we report the results for the regularisation strength $\lambda \in \{10, 100, 1000\}$. Smaller values did not cause a noticeable effect on fairness, while larger values caused a strong degradation in terms of AUC.

Along with the link prediction AUC score, all methods were tested for their deviation from Demographic Parity (DP) and Equalised Opportunity (EO). The calculation of those measures follows (Buyl & De Bie, 2020), where DP is the maximal difference between the mean predicted value of any subgroup. Similarly, the EO measure refers to the maximal difference between true positive rates of subgroups. Lower DP and EO scores therefore imply a fairer model. Note that the test set contains proportionally less negative edges than the overall dataset, possibly skewing the DP score. This effect was compensated for by proportionately increasing the contribution of negative samples when calculating DP. Furthermore, additional measures are reported in the Appendix on the diversity in the ranking of prediction scores, as well as diversity in the embeddings.

5.5. Results

The results of our experiments on the considered datasets are shown in Fig. 1. A table with the results in text format is provided in the Appendix.

Fairness Quality
In many cases and across all four methods, it can be seen that the use of our proposed fairness regularizer indeed significantly reduces the link prediction bias, according to the employed fairness criterions. This is in contrast to the baselines DEBAYES and CFC. The former did not improve fairness scores over CNE, while the latter could only become more fair at a significant cost to AUC.

There are a few exceptions where our method does not reduce unfairness according to the fairness criterion. First, there are some cases where an already low DP score for the base method can not be improved further by adding the DP regularizer. This happens for MAXENT in Fig. 1a, GAE and DOT-PRODUCT in Fig. 1b and for CNE in Fig. 1c. A second kind of exception is where the method with the DP regularizer is less EO-unfair than with the EO regularizer. It occurs for the DOT-PRODUCT (EO) variant in Fig. 1a and 1c, possibly because the former had a larger reduction in predictive power overall. In both these cases, DOT-PRODUCT (EO) still significantly reduces EO compared to the DOT-PRODUCT model without fairness regularizer.

Predictive Quality
Moreover, the decrease in AUC is fairly minimal with our fairness regularizer, especially compared to an adversarial approach like CFC. While the addition of the EO regularizer has no noticeable effect on the AUC, the DP variant does cause strong reduction on some models in Fig. 1a. This is to be expected, because enforcing DP can cause a significant loss in predictive power if the subgroups in the underlying data have different base rates (Hardt et al., 2016). For a network like POLBLOGS, which strongly favours intra-group connections, encouraging the inter-group connections therefore results in AUC loss.

Table 2: Median runtimes (s) of measured by Python's `time.perf_counter`.

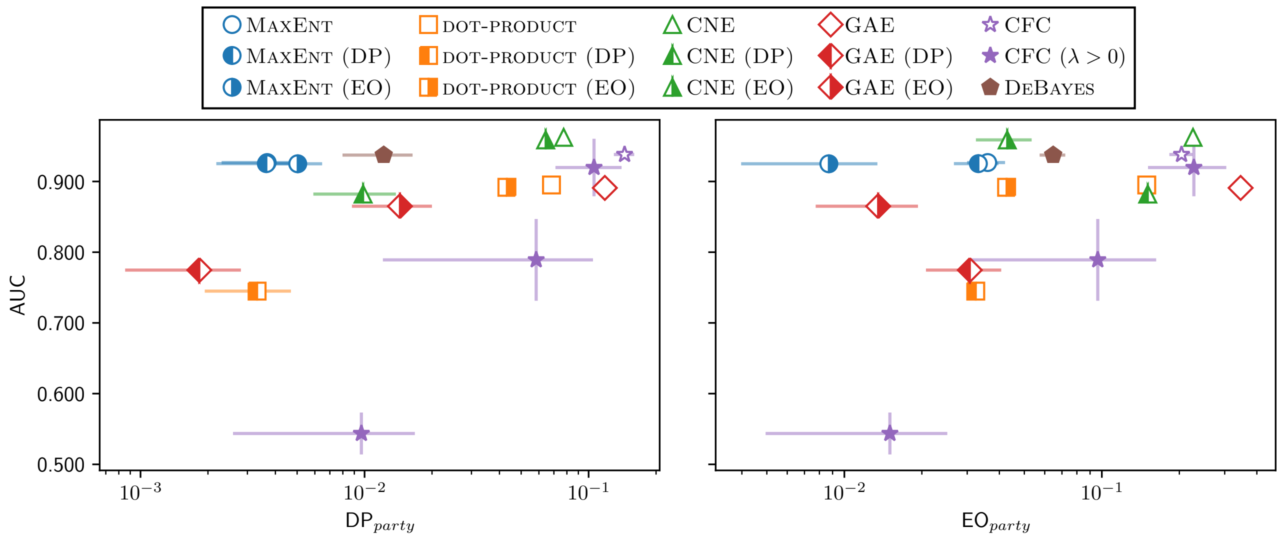
METHOD	POLBLOGS	ML100K	FACEBOOK
MAXENT	14	68	158
MAXENT (DP)	707	3050	1924
MAXENT (EO)	170	773	1191
DOT-PRODUCT	60	62	200
DOT-PRODUCT (DP)	349	456	1169
DOT-PRODUCT (EO)	135	239	531
CNE	105	307	349
CNE (DP)	574	1417	2065
CNE (EO)	286	843	865
GAE	28	26	101
GAE (DP)	278	437	1072
GAE (EO)	92	255	388
CFC	280	843	1601
CFC ($\lambda > 0$)	242	2623	3494
DEBAYES	98	305	343

Runtimes
Runtimes[2] of each method are listed in Tab. 2. The addition of our regularizer causes a large increase in runtime. The biggest reason for this is that we compute the I-projection, a full optimization, for every update of h. We already used some easy speed improvements, discussed in Sec. 4.2, but there is further research required into speeding up our implementation, e.g. by subsampling the model h or relaxing the convergence criteria.

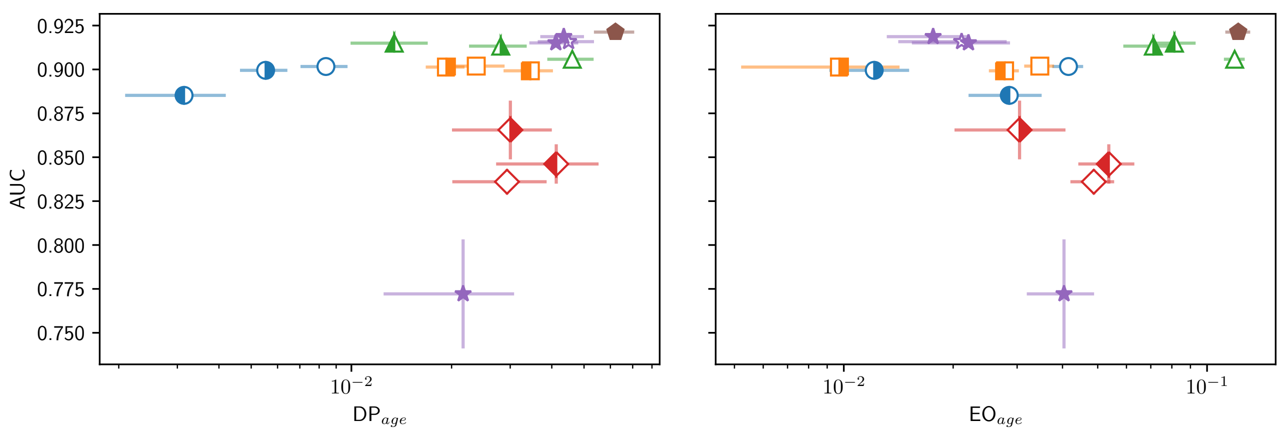
6. Conclusion

Employing a generic way to characterize the set of fair link prediction distributions, we can compute the I-projection of any link predictor model onto this set. That distance, i.e. the KL-divergence between the model and its I-projection, can then be used as a principled regularizer during the training process and can be applied to a wide range of fairness criteria. We evaluated the benefit of our proposed method for two such criteria: demographic parity and equalized opportunity. Overall, our regularizer caused significant improvements in the desired fairness notions, at a relatively minimal cost in predictive power. In this it outperformed the baseline fairness modifications for graph embedding methods, which could not leverage its debiased embeddings to perform fair link prediction according to generic fairness criteria. In the future, more task-specific link prediction fairness criteria can be defined within our framework, taking inspiration from social graph or recommender systems literature. Moreover, our proposed regularizer can be extended beyond graph data structures.

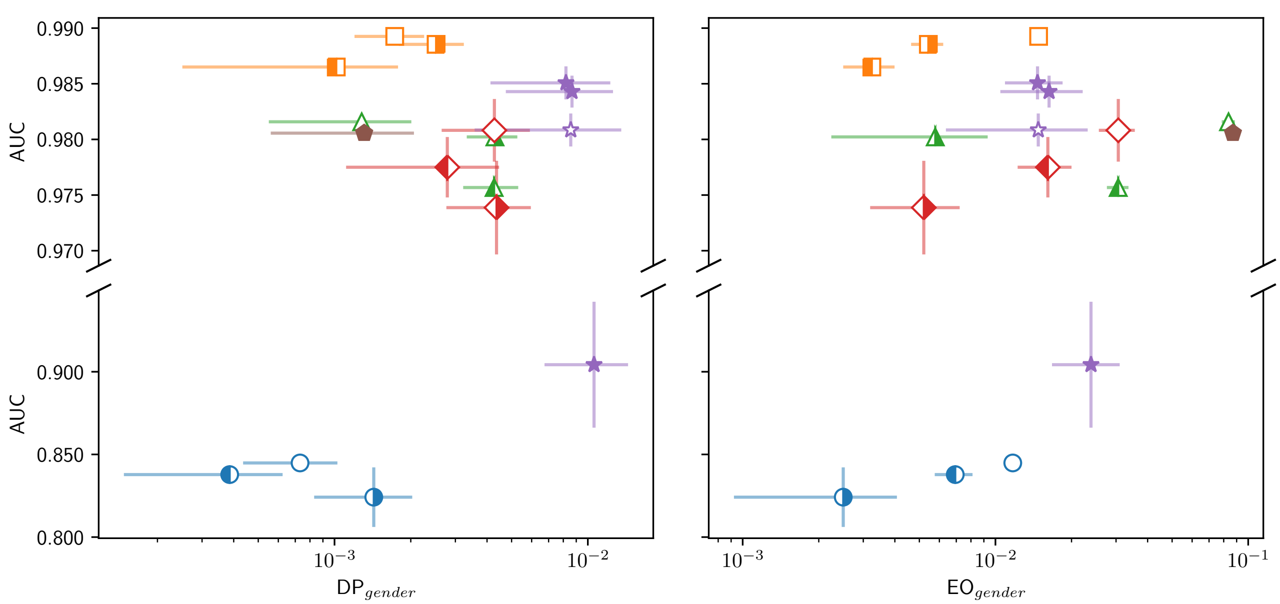
[2]All experiments were conducted using half the hyperthreads on a machine equipped with a 12 Core Intel(R) Xeon(R) Gold processor and 256GB of RAM



(a) Results on the POLBLOGS dataset.



(b) Results on the ML100K dataset.



(c) Results on the FACEBOOK dataset.

Figure 1: Markers display the mean over ten identical experiment runs with different random seeds. Error bars horizontally and vertically show the standard deviation. Completely empty markers refer to methods without any fairness modification. Methods with a fairness regularizer that enforces the DP or EO fairness criterion are left-filled or right-filled respectively. On the x-axis, unfairness is measured, so lower is better. On the y-axis, AUC is measured, so higher is better.

Acknowledgements

This research was funded by the ERC under the EU's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, the Flemish Government (AI Research Program), the BOF of Ghent University (PhD scholarship BOF20/DOC/144), and the FWO (project no. G091017N, G0F9816N, 3G042220).

References

Adamic, L. A. and Glance, N. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43, 2005.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. A reductions approach to fair classification. In *International Conference on Machine Learning*, pp. 60–69. PMLR, 2018.

Alghamdi, W., Asoodeh, S., Wang, H., Calmon, F. P., Wei, D., and Ramamurthy, K. N. Model projection: Theory and applications to fair machine learning. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2711–2716. IEEE, 2020.

Bose, A. and Hamilton, W. Compositional fairness constraints for graph embeddings. In *International Conference on Machine Learning*, pp. 715–724, 2019.

Burnham, K. P. and Anderson, D. R. Practical use of the information-theoretic approach. In *Model selection and inference*, pp. 75–117. Springer, 1998.

Buyl, M. and De Bie, T. Debayes: a bayesian method for debiasing network embeddings. In *International Conference on Machine Learning*, pp. 1220–1229. PMLR, 2020.

Calmon, F. P., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R. Optimized pre-processing for discrimination prevention. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3995–4004, 2017.

Cotter, A., Jiang, H., Gupta, M. R., Wang, S., Narayan, T., You, S., and Sridharan, K. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *Journal of Machine Learning Research*, 20(172):1–59, 2019.

Cover, T. M. *Elements of information theory*. John Wiley & Sons, 1999.

Csiszár, I. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Csiszár, I. and Matus, F. Information projections revisited. *IEEE Transactions on Information Theory*, 49(6):1474–1490, 2003.

De Bie, T. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. *Data Mining and Knowledge Discovery*, 23(3):407–446, 2011.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Hamilton, W. L., Ying, R., and Leskovec, J. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 3323–3331, 2016.

Hofstra, B., Corten, R., Van Tubergen, F., and Ellison, N. B. Sources of segregation in social networks: A novel approach using facebook. *American Sociological Review*, 82(3):625–656, 2017.

Jiang, H. and Nachum, O. Identifying and correcting label bias in machine learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 702–712. PMLR, 2020.

Kallus, N. and Zhou, A. The fairness of risk scores beyond classification: Bipartite ranking and the xauc metric. *Advances in neural information processing systems*, 32, 2019.

Kang, B., Lijffijt, J., and De Bie, T. Conditional network embeddings. In *International Conference on Learning Representations*, 2018.

Karimi, F., Génois, M., Wagner, C., Singer, P., and Strohmaier, M. Homophily influences ranking of minorities in social networks. *Scientific reports*, 8(1):1–12, 2018.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Liben-Nowell, D. and Kleinberg, J. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7): 1019–1031, 2007.

Martínez, V., Berzal, F., and Cubero, J.-C. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, 49(4):1–33, 2016.

Masrour, F., Wilson, T., Yan, H., Tan, P.-N., and Esfahanian, A. Bursting the filter bubble: Fairness-aware network link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 841–848, 2020.

McAuley, J. and Leskovec, J. Learning to discover social circles in ego networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 539–547, 2012.

McPherson, M., Smith-Lovin, L., and Cook, J. M. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

Rahman, T., Surma, B., Backes, M., and Zhang, Y. Fairwalk: Towards fair graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 3289–3295. International Joint Conferences on Artificial Intelligence Organization, 2019.

Robins, G., Pattison, P., Kalish, Y., and Lusher, D. An introduction to exponential random graph (p*) models for social networks. *Social networks*, 29(2):173–191, 2007.

Wei, D., Ramamurthy, K. N., and Calmon, F. Optimized score transformation for fair classification. In *International Conference on Artificial Intelligence and Statistics*, pp. 1673–1683. PMLR, 2020.

Woodworth, B., Gunasekar, S., Ohannessian, M. I., and Srebro, N. Learning non-discriminatory predictors. In *Conference on Learning Theory*, pp. 1920–1953. PMLR, 2017.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pp. 962–970. PMLR, 2017.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 5171–5181, 2018.

A. Hyperparameters and Implementation

As stated in the paper, hyperparameter tuning was minimal. In this section, we nevertheless provide, for each method, additional details on the choice of hyperparameters and implementations. The source code is also provided in the supplementary submission.

MAXENT
The MAXENT model is used in both the CNE and DEBAYES models as a prior distribution. We therefore used the code available for CNE (available here) as a starting point for developing a *PyTorch* version of this model. The optimisation was done with *LBFGS*, with at most 100 iterations.

DOT-PRODUCT
Due to the simplicity of the DOT-PRODUCT model, it was implemented from scratch in *PyTorch*. A dimensionality of 128 performed the best among the considered values $\{8, 16, 128\}$. The model was optimised with *Adam* and at a learning rate of 0.01 with 100 iterations.

CNE
The CNE implementation was based on the code from DEBAYES, since the latter is a fairness adaptation of CNE. Our *PyTorch* modification was made similar to DOT-PRODUCT, except that it uses the distance between node embeddings instead of the actual dot product, and it uses MAXENT as a prior distribution. As hyperparameters, we used the default values as listed for DEBAYES: dimensionality of 8, learning rate of 0.1 and s_2 value of 16. However, we found the method already converged with 200 epochs, as opposed to the default 1000.

GAE
For the GAE implementation, we closely followed the source code here. We used similar parameters as DOT-PRODUCT, but found a dimensionality of 16 achieved better validation scores.

CFC
We applied the CFC method to link prediction by using it as a dot-product decoder. As such, the hyperparameters are the same as for DOT-PRODUCT, though we stayed as close as possible to the original implementation available here.

DEBAYES
Our implementation of DEBAYES is based on the publicly available version here. However, we used our own *PyTorch* version of CNE in order to compare both methods easily. The DEBAYES method has no other hyperparameters besides those listed for CNE.

B. Additional Measures

In addition to the DP and EO measures, we evaluated the considered methods on two additional fairness measures.

Rank DP (RDP)
As stated in the paper, we computed the DP measure as the maximum difference between the mean link prediction probability of each subgroup combination, proposed in prior work (Buyl & De Bie, 2020). A possible drawback of this approach is that an algorithm can reduce its DP score by simply making all prediction probabilities very similar. For example, a dot-product decoder may give privileged vertex pairs a link probability of 0.9, and unprivileged pairs a score of 0.5. Such a method would have a DP score of 0.4. However, e.g. by rescaling the embeddings, a similar dot-product decoder may predict scores of 0.6 and 0.5 for privileged and unprivileged pairs respectively, which would result in a DP score of 0.1. Both methods provide total separation in the *ranks* of the scores between subgroups, so it could be argued that they are both extremely (yet equally) discriminatory.

Taking inspiration from prior work (Kallus & Zhou, 2019), consider the AUC score, computed over the link prediction scores on the test set, when pairs of vertices of sensitive groups V_s and V_t respectively are the positive class and vertices with a different sensitive attribute combination are the negative class. Let the *Rank Demographic Parity* (RDP) score be the maximum of those AUC scores, over all sensitive value pairs $(s, t) \in S$. Clearly, when the prediction scores are completely separated between groups as in the earlier example, the RDP score would equal 1.

In the results reported in Sec. C, the RDP score heavily correlated with the DP score, suggesting that the latter is already an adequate measure of demographic parity in our experiment setup.

Representation Bias (RB)
The baseline methods DEBAYES (Buyl & De Bie, 2020) and CFC (Bose & Hamilton, 2019) focus on making the embeddings themselves less biased. While this was not the direct aim of our fairness regularizer, we evaluate all considered methods by also measuring the *Representation Bias* (RB) (Buyl & De Bie, 2020), i.e. the maximum AUC score that a logistic regression classifier achieves when it attempts to predict the sensitive attribute value of a vertex' embedding.

We report the scores using such an evaluation measure in Sec. C and validate that DEBAYES and CFC indeed succeed in debiasing their embeddings, as reported in their respective papers. The hyperparameters of the classifier are tuned on a validation set of vertices among all vertices in the graph. In some cases, e.g. for DOT-PRODUCT (DP), the optimal hyperparameters are those that cause extreme regularisation, causing the RB scores to be 0.5.

C. Results in Table Format

The full experimental results, which were displayed as AUC-fairness trade-offs in the paper, are again displayed in Tab. 3, 4 and 5. In addition to the DP and EO fairness measures, the RDP and RB scores as described in Sec. B are also listed.

Table 3: Mean ± standard deviation of results on the POLBLOGS dataset.

METHOD	AUC	DP	EO	RDP	RB
CFC	0.938 ± 0.0047	0.145 ± 0.0151	0.204 ± 0.0211	0.664 ± 0.0154	0.977 ± 0.0049
CFC ($\lambda = 10$)	0.920 ± 0.0407	0.106 ± 0.0346	0.228 ± 0.0766	0.668 ± 0.0336	0.940 ± 0.1194
CFC ($\lambda = 100$)	0.789 ± 0.0578	0.058 ± 0.0463	0.096 ± 0.0662	0.626 ± 0.0442	0.859 ± 0.1188
CFC ($\lambda = 1000$)	0.544 ± 0.0297	0.010 ± 0.0071	0.015 ± 0.0101	0.520 ± 0.0135	0.595 ± 0.0482
CNE	0.962 ± 0.0022	0.077 ± 0.0033	0.226 ± 0.0079	0.679 ± 0.0063	0.979 ± 0.0095
CNE (DP)	0.882 ± 0.0045	0.010 ± 0.0039	0.151 ± 0.0080	0.577 ± 0.0058	0.765 ± 0.0284
CNE (EO)	0.959 ± 0.0022	0.064 ± 0.0027	0.043 ± 0.0105	0.651 ± 0.0094	0.971 ± 0.0102
DEBAYES	0.937 ± 0.0026	0.012 ± 0.0042	0.065 ± 0.0074	0.615 ± 0.0067	0.699 ± 0.0524
DOT-PRODUCT	0.895 ± 0.0035	0.068 ± 0.0021	0.149 ± 0.0056	0.738 ± 0.0070	0.975 ± 0.0062
DOT-PRODUCT (DP)	0.745 ± 0.0046	0.003 ± 0.0014	0.032 ± 0.0016	0.570 ± 0.0083	0.500 ± 0.0000
DOT-PRODUCT (EO)	0.892 ± 0.0037	0.043 ± 0.0016	0.043 ± 0.0034	0.657 ± 0.0076	0.946 ± 0.0069
GAE	0.891 ± 0.0031	0.118 ± 0.0035	0.346 ± 0.0309	0.753 ± 0.0138	0.983 ± 0.0061
GAE (DP)	0.775 ± 0.0091	0.002 ± 0.0010	0.031 ± 0.0100	0.583 ± 0.0143	0.825 ± 0.0419
GAE (EO)	0.865 ± 0.0135	0.014 ± 0.0056	0.014 ± 0.0058	0.648 ± 0.0194	0.972 ± 0.0061
MAXENT	0.927 ± 0.0026	0.004 ± 0.0014	0.036 ± 0.0061	0.617 ± 0.0070	/
MAXENT (DP)	0.925 ± 0.0028	0.004 ± 0.0015	0.033 ± 0.0065	0.617 ± 0.0084	/
MAXENT (EO)	0.925 ± 0.0038	0.005 ± 0.0014	0.009 ± 0.0047	0.604 ± 0.0042	/

Table 4: Mean ± standard deviation of results on the ML100K dataset.

METHOD	AUC	DP	EO	RDP	RB
CFC	0.916 ± 0.0021	0.045 ± 0.0086	0.021 ± 0.0070	0.531 ± 0.0040	0.631 ± 0.0191
CFC ($\lambda = 10$)	0.919 ± 0.0021	0.043 ± 0.0065	0.018 ± 0.0045	0.530 ± 0.0030	0.662 ± 0.0127
CFC ($\lambda = 100$)	0.915 ± 0.0023	0.041 ± 0.0069	0.022 ± 0.0066	0.530 ± 0.0036	0.681 ± 0.0158
CFC ($\lambda = 1000$)	0.772 ± 0.0311	0.022 ± 0.0092	0.040 ± 0.0085	0.519 ± 0.0035	0.547 ± 0.0235
CNE	0.906 ± 0.0015	0.046 ± 0.0073	0.119 ± 0.0078	0.536 ± 0.0030	0.547 ± 0.0123
CNE (DP)	0.915 ± 0.0013	0.013 ± 0.0035	0.081 ± 0.0117	0.511 ± 0.0020	0.588 ± 0.0205
CNE (EO)	0.913 ± 0.0014	0.028 ± 0.0055	0.071 ± 0.0124	0.528 ± 0.0032	0.614 ± 0.0134
DEBAYES	0.921 ± 0.0013	0.062 ± 0.0086	0.122 ± 0.0096	0.536 ± 0.0028	0.524 ± 0.0235
DOT-PRODUCT	0.902 ± 0.0012	0.024 ± 0.0051	0.035 ± 0.0033	0.528 ± 0.0029	0.657 ± 0.0233
DOT-PRODUCT (DP)	0.899 ± 0.0012	0.034 ± 0.0058	0.028 ± 0.0026	0.509 ± 0.0037	0.759 ± 0.0232
DOT-PRODUCT (EO)	0.901 ± 0.0012	0.019 ± 0.0026	0.010 ± 0.0045	0.526 ± 0.0029	0.697 ± 0.0275
GAE	0.836 ± 0.0072	0.029 ± 0.0092	0.049 ± 0.0067	0.518 ± 0.0040	0.582 ± 0.0279
GAE (DP)	0.846 ± 0.0112	0.041 ± 0.0140	0.054 ± 0.0094	0.510 ± 0.0050	0.645 ± 0.0195
GAE (EO)	0.866 ± 0.0167	0.030 ± 0.0100	0.030 ± 0.0103	0.517 ± 0.0034	0.578 ± 0.0325
MAXENT	0.902 ± 0.0014	0.008 ± 0.0014	0.042 ± 0.0040	0.536 ± 0.0026	/
MAXENT (DP)	0.885 ± 0.0039	0.003 ± 0.0011	0.029 ± 0.0065	0.515 ± 0.0055	/
MAXENT (EO)	0.899 ± 0.0013	0.006 ± 0.0009	0.012 ± 0.0030	0.530 ± 0.0018	/

Table 5: Mean ± standard deviation of results on the FACEBOOK dataset.

METHOD	AUC	DP	EO	RDP	RB
CFC	0.981 ± 0.0015	0.009 ± 0.0050	0.015 ± 0.0084	0.529 ± 0.0068	0.601 ± 0.0150
CFC ($\lambda = 10$)	0.984 ± 0.0014	0.009 ± 0.0039	0.016 ± 0.0058	0.529 ± 0.0040	0.615 ± 0.0153
CFC ($\lambda = 100$)	0.985 ± 0.0015	0.008 ± 0.0041	0.015 ± 0.0038	0.529 ± 0.0049	0.617 ± 0.0157
CFC ($\lambda = 1000$)	0.904 ± 0.0381	0.011 ± 0.0038	0.024 ± 0.0071	0.532 ± 0.0052	0.586 ± 0.0228
CNE	0.982 ± 0.0006	0.001 ± 0.0007	0.084 ± 0.0050	0.538 ± 0.0035	0.601 ± 0.0189
CNE (DP)	0.976 ± 0.0006	0.004 ± 0.0010	0.031 ± 0.0030	0.512 ± 0.0027	0.594 ± 0.0217
CNE (EO)	0.980 ± 0.0006	0.004 ± 0.0010	0.006 ± 0.0035	0.518 ± 0.0025	0.607 ± 0.0192
DEBAYES	0.981 ± 0.0004	0.001 ± 0.0007	0.087 ± 0.0042	0.539 ± 0.0030	0.592 ± 0.0163
DOT-PRODUCT	0.989 ± 0.0006	0.002 ± 0.0005	0.015 ± 0.0007	0.561 ± 0.0037	0.630 ± 0.0138
DOT-PRODUCT (DP)	0.987 ± 0.0008	0.001 ± 0.0008	0.003 ± 0.0007	0.515 ± 0.0032	0.500 ± 0.0000
DOT-PRODUCT (EO)	0.989 ± 0.0005	0.003 ± 0.0007	0.005 ± 0.0008	0.530 ± 0.0041	0.769 ± 0.0185
GAE	0.981 ± 0.0028	0.004 ± 0.0016	0.031 ± 0.0050	0.545 ± 0.0037	0.602 ± 0.0152
GAE (DP)	0.977 ± 0.0027	0.003 ± 0.0017	0.016 ± 0.0039	0.518 ± 0.0061	0.626 ± 0.0193
GAE (EO)	0.974 ± 0.0042	0.004 ± 0.0016	0.005 ± 0.0020	0.535 ± 0.0026	0.618 ± 0.0240
MAXENT	0.845 ± 0.0014	0.001 ± 0.0003	0.012 ± 0.0007	0.541 ± 0.0041	/
MAXENT (DP)	0.838 ± 0.0043	0.000 ± 0.0002	0.007 ± 0.0012	0.530 ± 0.0051	/
MAXENT (EO)	0.824 ± 0.0180	0.001 ± 0.0006	0.002 ± 0.0016	0.519 ± 0.0070	/