DeBayes: a Bayesian Method for Debiasing Network Embeddings

Maarten Buyl [1] **Tijl De Bie** [1]

Abstract

As machine learning algorithms are increasingly deployed for high-impact automated decision making, ethical and increasingly also legal standards demand that they treat all individuals fairly, without discrimination based on their age, gender, race or other sensitive traits. In recent years much progress has been made on ensuring fairness and reducing bias in standard machine learning settings. Yet, for network embedding, with applications in vulnerable domains ranging from social network analysis to recommender systems, current options remain limited both in number and performance. We thus propose DeBayes: a conceptually elegant Bayesian method that is capable of learning *debiased* embeddings by using a *biased* prior. Our experiments show that these representations can then be used to perform link prediction that is significantly more fair in terms of popular metrics such as *demographic parity* and *equalized opportunity*.

1. Introduction

In recent years, machine learning algorithms have increasingly taken up decision making responsibilities in the real world. However, their traditional focus on accuracy makes them prone to discriminate in ways that are morally or legally unacceptable. This development has sparked a surge of research interest in machine learning *fairness* (Friedler et al., 2019), where the objective is to learn how to make decisions that remain accurate while avoiding to discriminate against individuals with respect to a particular set of *sensitive attributes*.

A naive attempt at preventing discrimination is to simply remove the sensitive information from the input, thereby avoiding what is seen as *disparate treatment*. Yet moral standards and anti-discrimination laws are typically also concerned about *disparate impact* (Barocas & Selbst, 2016). This alternative notion of discrimination demands that outcomes do not disproportionately hurt or benefit people with specific sensitive attributes (Zafar et al., 2017). In machine learning, avoiding disparate impact is often difficult, since the sensitive attributes are typically not independent of other features in the data (Pedreshi et al., 2008). f While research in fair binary classification has been plentiful, fairness has seen relatively little attention in the field of *network embedding* (Mehrabi et al., 2019). There, the goal is to find a low-dimensional representation for every node in a graph dataset, in order to perform various downstream tasks such as node classification, community detection, and link prediction (Cui et al., 2018). The latter is concerned with estimating, given the observed network structure, the likelihood of an edge between two nodes. The field has seen tremendous interest in recent years due to the ubiquity of the graph structured data, with applications in e.g. social graphs and recommender systems (Hamilton et al., 2017). However, these networks often represent connections between people, possibly giving rise to fairness issues. For example, friendship recommendation in social media graphs may reinforce historical biases towards certain groups (Stoica et al., 2018). A similar issue has been found in book recommendation, where algorithms may be biased towards suggesting books with male authors (Ekstrand et al., 2018).

In this work, our aim is to provide an elegant method to increase fairness at both the level of network embedding and at the level of its downstream task: link prediction. An algorithm that naturally allows us to do so is *Conditional Network Embedding* (CNE) (Kang et al., 2019): a Bayesian network embedding algorithm that uses a *prior distribution* to model the structural properties of the network, such as the node degrees or any block structure. Given this, CNE then searches for an embedding that optimally describes the affinities between nodes *insofar these are not already captured by the prior*.

Proposed method We introduce *DeBayes*, an adaptation of CNE where the sensitive information is modelled as strongly as possible in the prior distribution. Consequently, the learned embeddings no longer need to represent this information, i.e. it is debiased. The idea is illustrated with

---

[1]Department of Electronics and Information Systems (ELIS), IDLab, Ghent University, Ghent, Belgium. Correspondence to: Maarten Buyl <maarten.buyl@ugent.be>, Tijl De Bie <tijl.debie@ugent.be>.

(a) oblivious training prior   (b) biased training prior
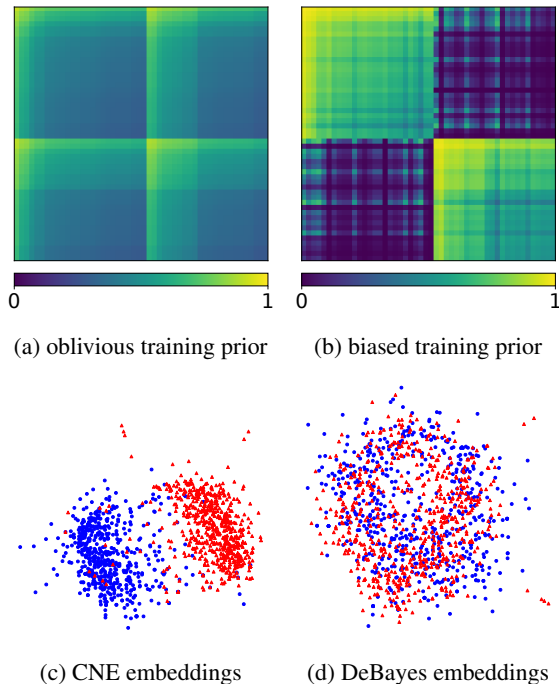

(c) CNE embeddings   (d) DeBayes embeddings

Figure 1. An illustration of DeBayes' main idea. The nodes represent political blogs on US politics and they have an edge if one of them hyperlinks to the other. The sensitive attribute is whether the blog author leans liberal (blue circles) or conservative (red triangles). The matrices (a, b) display, for a link between blogs i and j, the prior probability at location (i, j) as a heatmap. Only the 50 highest degree nodes are shown, grouped by political leaning and sorted by degree. The scatter plots (c,d) show the 2-dimensional embeddings that CNE learns if the respective prior is used.

a 2-dimensional embedding of the *political blogs* network (Adamic & Glance, 2005) in Figure 1. Panel (a) shows an *oblivious* prior probability matrix, which takes the observed node degrees into account, but is oblivious to the sensitive attribute (political leaning). Nodes are grouped per political leaning (liberal and conservative), and sorted in order of decreasing degree within each group. Panel (c) shows the corresponding embedding found by CNE, clearly showing a separation between both groups, as it needs to represent the fact that same-leaning blogs are connected more frequently. Panels (b) and (d) on the other hand show a biased prior (edges between groups are less probable than within) and the matching CNE embedding. Here no separation is required as the prior already represents the bias.

Of course, link predictions with a CNE embedding and matching prior will discriminate regardless of the prior: if the prior does not represent the bias, then the embedding will. DeBayes' strategy, however, is to *train* the CNE embedding using a biased prior, resulting in debiased embedding, but then to *evaluate* the probability of links (i.e. do link prediction) in combination with the oblivious prior, as illus-
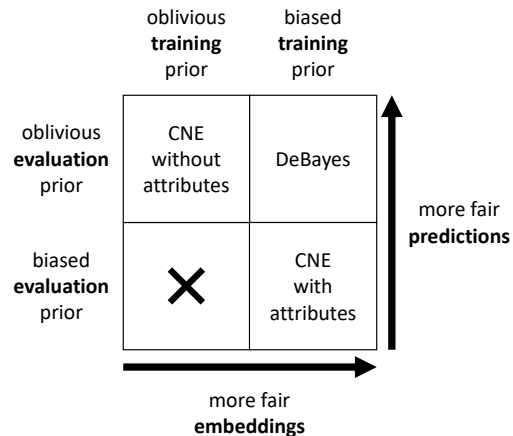

Figure 2. The results of different combinations of oblivious or biased priors at training and evaluation time. Note that the combination of 'oblivious training prior' and 'biased evaluation prior' is rarely desirable, as it enforces unfairness while likely decreasing prediction accuracy.

trated in Figure 2. We will show that this strategy reduces bias, while keeping link prediction accuracy high.

DeBayes has several conceptual advantages. First, it is simple to choose whether fairness is applied or not at evaluation time; it suffices to choose the required evaluation prior. Second, the prior is flexible enough to model several sensitive attributes, such as gender and age, at the same time. Third, imposing fairness in this way implies no additional computational cost. Finally, the concept behind DeBayes could be extended to other fields within representation learning.

Outline The rest of the paper is organized as follows. Section 2 briefly surveys related work. Section 3 proposes a measure of what is meant by 'bias in embeddings', and also extends the well-known fairness definitions *demographic parity* and *equalized opportunity* to link prediction. Section 4 discusses how the priors of DeBayes are constructed and how CNE is adapted to generate debiased embeddings and link predictions. Section 5 empirically confirms DeBayes' superiority when compared with state-of-the-art baselines. Section 6 offers conclusions and a forward look.

2. Related Work

Broadly speaking, the fairness of a machine learning model can be improved in one of three stages in its pipeline (Friedler et al., 2019): in preprocessing, in the algorithm itself or in postprocessing. The *preprocessing* techniques focus on modifying the training data in order to reduce the predictability of its sensitive attributes (Feldman et al., 2015; Calmon et al., 2017). Conversely, *postprocessing* methods try to make a discriminatory predictor fair e.g. by correcting its output probabilities such that a specific fairness con-

straint is satisfied (Hardt et al., 2016). However, depending on only such a post-hoc correction to achieve fairness can significantly hurt the accuracy of the predictor (Woodworth et al., 2017), making it often unpractical.

DeBayes belongs to the category of *algorithmic* fairness. Such methods usually try to prevent bias from affecting the learned model. A common approach is to modify the loss function, e.g. by adding fairness-inducing regularisation terms (Kamishima et al., 2012) or by requiring that fairness constraints are met throughout the learning process (Zafar et al., 2017; Woodworth et al., 2017). Furthermore, (Calders & Verwer, 2010) introduced naive Bayes approaches where it is assumed that latent, non-discriminatory variables can be learned that model a fair dependency between the ground truth and the data. There, a prior distribution was used to aid in finding those latent variables. Our method makes comparable assumptions, with embeddings as latent variables.

In general, the strategy of learning latent, non-discriminatory embeddings is referred to as *fair representation learning*. The goal is to represent entities in the input data as informative embeddings in a metric space, without incorporating sensitive information. A popular approach is to penalize bias in the embeddings using an additional loss term. For example, this term can explicitly measure the divergence from a fairness requirement (Zemel et al., 2013) or can be the negative loss function of an adversarial classifier that attempts to predict sensitive attributes from the embeddings (Madras et al., 2018). Of particular relevance is the Flexibly Fair Variational Autoencoder (Creager et al., 2019), which enforces disentanglement between sensitive and non-sensitive information in embeddings during training. During evaluation, sensitive embedding factors can be removed, thereby avoiding their contribution in downstream tasks. The DeBayes method also attempts to represent sensitive and non-sensitive information separately during training, such that the bias can be removed afterwards. However, DeBayes differs from this prior work in that the bias is avoided through the Bayesian formulation of the objective function, rather than by reducing bias through a loss term. Consequently, bias is avoided insofar as it is present in the prior distribution, not through a manually chosen hyperparameter in the loss function. Moreover, DeBayes can incorporate any bias that can be defined through probabilities, even aggregated sensitive information.

Our interest is specifically in providing fairness for *network* embedding algorithms and their link predictions. In our experiments, we use two baselines from the field of fair network embedding. The first is Fairwalk (Rahman et al., 2019): an adaptation of the popular network embedding method *node2vec* (Grover & Leskovec, 2016). It aims to address fairness issues by weighing the neighbourhood of each node in favour of under-represented groups. The effect

is that the random walks of node2vec will be less discriminatory, with the expectation that downstream tasks will be more fair as well. The second baseline is *Compositional Fairness Constraints* (Bose & Hamilton, 2019): an adversarial approach where the encoder is trained to produce embeddings for which its adversaries can not predict the sensitive attributes.

3. Fairness in Link Prediction

The research in fair machine learning has given rise to several different (and sometimes conflicting) definitions of fairness. However, they are usually specialised to binary classification. In this section, we therefore extend these notions of fairness to link prediction. After introducing some notation in Section 3.1, we consider interpretations of fairness at the level of representation learning in Section 3.2 and the higher level of link prediction in 3.3. These can then be used to inspire and evaluate the fairness properties of our method.

3.1. Notation

Let $G = (V, E, A)$ be an undirected, attributed graph with set of nodes V, set of edges $E \subseteq V \times V$ and node attributes A. Without loss of generality for DeBayes (see Section 4.4), we first consider the case where every node has exactly one attribute, which is somehow sensitive, with value $s \in S$. The mapping $A : V \mapsto S$ then returns the sensitive attribute value for every node. For example, if *gender* is the type of sensitive information and v is a node for a male user, then $A(v) = $ 'male'. Below we will use $\hat{G} = (V, \hat{E}, \hat{A})$ to explicitly refer to empirically observed variables.

As mentioned before, network embedding methods generally attempt to map every node $v \in V$ to a d-dimensional vector space $\mathbb{R}^d$ in a way that supports network inference tasks. Let $\mathbf{z}_v \in \mathbb{R}^d$ be the embedding vector for node v. Our interest is particularly in the task of link prediction, concerned with estimating the probability $P_{u,v}$ that there exists a link (u, v) between nodes u and v.

3.2. Fair Representations

In fair representation learning, the aim is to find the embedding $\mathbf{z}_v$ that best encodes the information about the node v, while its sensitive attribute value $s = A(v)$ is approximately obfuscated (Zemel et al., 2013). The underlying idea is that if the embedding is well-obfuscated, i.e. contains no sensitive information, then those embeddings can be used in a black-box fashion to achieve fairness in downstream tasks. Building upon previous work (Bose & Hamilton, 2019; Palowitch & Perozzi, 2019), we define an embedding as obfuscated if a classifier, trained on the representations and attributes of a subset of the nodes, has a low prediction accuracy on the unseen nodes. If the accuracy is equal to that of a

random classifier, then the embeddings are well-obfuscated and thus the **representation bias (RB)** is removed.

For a classifier c, let $P_c(s, \mathbf{z}_v)$ be its estimated probability (or any other score) that node v with embedding $\mathbf{z}_v$ has sensitive attribute value s. To measure its prediction performance, we chose the weighted one-versus-rest AUC score[1]. By taking the *weigthed* average over the one-versus-rest AUC scores, the contribution of the set $V_s = \{v \mid A(v) = s\}$, i.e. the nodes with sensitive value s, is proportional to the size of their group $|V_s|$. The RB score is then defined as:

$$\text{RB} = \sum_{s \in S} \frac{1}{|V_s|} \text{AUC}(\{P_c(A(v) \mid \mathbf{z}_v) \mid \forall v \in V_s\}).$$

While unbiased embeddings can be a useful component for a fair link prediction algorithm, limiting the fairness analysis of a link prediction method to only its generated node embeddings is not sufficient. For real-world applications, the main concern may only be for the fairness properties of the predictions themselves, not the embeddings. The specific case of CNE also presents the practical issue that it uses both embeddings and a prior to generate predictions. It may therefore still provide unfair link prediction, even when its intermediate result, the embeddings, are completely unbiased. More generally, link prediction algorithms like recommender systems may not use embeddings at all.

3.3. High-Level Fairness Measures for Link Prediction

When analysing fairness at the level of link prediction, we consider high-level measures that treat the method as a black box. To this end, we directly draw inspiration from two popular measures from the literature on fair binary classification: *demographic parity* and *equalized opportunity* (Hardt et al., 2016). In addition to introducing these measures, we also consider *acceptance rate parity* in our experiments.

Demographic parity The definition of fairness that most closely aligns with the notion of disparate impact is *demographic parity* (DP). It requires that predictions are independent of the sensitive attribute. Applied to the task of link prediction, we note that both ends may have sensitive attributes in a link. Therefore, the prediction probabilities should be independent from the attributes of both nodes in the link. As such, we define DP to be achieved when the prediction rates $\delta_{a,b}$ are equal for every pair of sensitive attribute values a and b. Let $N_{a,b} = |\{(u, v) \mid u \in V_a, v \in V_b\}|$, i.e. all possible combinations between nodes with attribute values a and b. The prediction rates $\delta_{a,b}$ are then computed as:

$$\delta_{a,b} = \frac{1}{N_{a,b}} \sum_{u \in V_a} \sum_{v \in V_b} P_{u,v}.$$

For example, an undirected social network where the sensitive attribute is $s \in \{M, F\}$ would have rates $\delta_{M,F}$, $\delta_{M,M}$ and $\delta_{F,F}$ be equal[2]. However, trying to achieve any fairness notion will usually incur a trade-off between model accuracy and fairness (Friedler et al., 2019). To quantify the deviation from this 'perfect' DP, we define the measure as the maximum absolute difference (Woodworth et al., 2017) between any pair of values $\delta_{a,b}$ and $\delta_{c,d}$:

$$\text{DP} = \max_{a,b,c,d} |\delta_{a,b} - \delta_{c,d}|.$$

A crucial drawback of the concept behind demographic parity is that it may cripple the prediction quality (Hardt et al., 2016). For example, if M-valued users in a social graph are more often interconnected in general, then an ideal predictor that perfectly predicts the target output, will have a higher $\delta_{M,M}$ than the other rates. In some applications, such an ideal predictor is not seen as discriminatory because it simply reflects the ground truth.

Equalized opportunity A different interpretation of fairness was therefore proposed: *equalized opportunity* (EO). It asserts independence between the true positive rate and the sensitive attribute. Let $N_{a,b}^+ = |\{(u, v) \mid u \in V_a, v \in V_b, (u, v) \in E\}|$, i.e. all the edges E of which the ends have attribute value a and b respectively. The true positive rate for a certain attribute pair is then $\epsilon_{a,b}$:

$$\epsilon_{a,b} = \frac{1}{N_{a,b}^+} \sum_{u \in V_a} \sum_{v \in V_b} P_{u,v} \, \mathbb{1}((u, v) \in E).$$

where $\mathbb{1}(\cdot)$ represents the identity function. Similarly to DP, we define the measure to compare EO quantitatively as:

$$\text{EO} = \max_{a,b,c,d} |\epsilon_{a,b} - \epsilon_{c,d}|.$$

Acceptance rate parity Finally, note that other fairness measures have been introduced for link prediction. One example is a form of demographic parity based on *acceptance rates* (Rahman et al., 2019), which we will also consider in our analysis. The acceptance rate $\alpha_{a,b}$ measures the relative frequency at which a link with attributes a and b appears in the overall top-k% highest predictions scores. When a link between nodes u and v is among those top scores, it is said that v is part of the recommendations $\rho(u)$ of node u:

$$\alpha_{a,b} = \frac{1}{N_{a,b}} |\{(u, v) \mid v \in \rho(u), u \in V_a, v \in V_b\}|.$$

We refer to this fairness notion as *acceptance rate parity* (ARP). It is measured as the variance between rates:

$$\text{ARP} = \text{Var}(\{\alpha_{a,b} \mid a \in S, b \in S\}).$$

---

[1]In our experiments, we used the `roc_auc_score` implementation from the `scikit-learn 0.22.1` library.

[2]In this undirected network, $\delta_{M,F} = \delta_{F,M}$ already holds.

4. The DeBayes Method

Our goal for algorithmic fairness is to reduce the impact of the sensitive attribute on the algorithm's learning process while maintaining accuracy in its predictions. To this end, we propose DeBayes, a method that generates debiased network embeddings by modelling the sensitive information in a prior distribution.

Recall that DeBayes is based on *Conditional Network Embedding* (CNE) (Kang et al., 2019), the principles of which will be briefly summarized in Section 4.1. As illustrated in Figure 2, DeBayes extends CNE by utilizing a biased prior for *training* and an oblivious prior during *evaluation*. Their construction is discussed in Section 4.2 and Section 4.3. Finally, some generalizations are made in Section 4.4

4.1. Conditional Network Embedding

CNE finds an embedding $\mathbf{Z}$ using Maximum Likelihood estimation given a model $P(\hat{G}|\mathbf{Z})$, with $\hat{G}$ the empirical graph. Key to CNE is that the likelihood function is formulated using Bayes rule: $P(\hat{G}|\mathbf{Z}) \triangleq \frac{P(\mathbf{Z}|\hat{G}) \cdot P(\hat{G})}{P(\mathbf{Z})}$. Here, the prior models properties of the graph that are hard to represent by an embedding, such as the node degrees, or any known block structure in the graph (see below), while the postulated form (Kang et al., 2019) of the embedding's conditional probability $P(\mathbf{Z}|\hat{G})$ assigns high probabilities to embeddings where connected nodes are embedded nearby, and disconnected nodes are further from each other. By modelling certain properties using the prior, the embedding itself can thus focus on representing other information. Interestingly, $P(\hat{G}|\mathbf{Z})$ thus computed factorizes over the node pairs, such that it can immediately be used for link prediction without the need for an edge embedding or additional training.

Prior knowledge about the graph can be quantified as statistics over its structure, e.g. total density of the graph, individual node degrees, any known block structure. The prior $P(G)$ can then be constructed as the *maximum entropy distribution* where the expected values of these statistics match the actual values over G. For example, for the degree of a particular node u, the prior should satisfy the following constraint where P represents the prior distribution:

$$\sum_{v \in V} P((u,v) \in E) = \sum_{v \in V} \mathbb{1}\left((u,v) \in \hat{E}\right). \quad (1)$$

In such a prior, $P((u,v) \in E)$ is high between any node pair v and u if their degrees are high. Thus, high degree nodes that are connected need not be embedded very nearby in the embedding space, since their connection is already explained by the general popularity of node u. Conversely, if both nodes u and v have low node degrees, then a connection between them is a priori improbable, and they will have to

be embedded closely in the embedding space.

4.2. Training with a Biased Prior

In DeBayes, the prior can be used to model structural information *including any bias* in the networks connectivity. Specifically, we define constraints on the expected value of statistics defined in terms of the sensitive attributes of the nodes. As the prior distribution is thus dependent on the sensitive attributes A, we will denote it as $P(G \mid A)$ and refer to it as a *biased* prior distribution. More specifically, we can enforce the constraint from Eq. (1) within each sensitive attribute block, requiring that for every node u, the expected number of neighbours with a particular attribute value s is equal to its empirical value. Formally, $\forall u \in V$ and $\forall s \in S$:

$$\sum_{v \in V_s} P((u,v) \in E \mid A(v) = s) = \sum_{v \in V_s} \mathbb{1}\left((u,v) \in \hat{E}\right). \quad (2)$$

Consider again an example from a social graph: if user u tends to mainly be connected to female users in the given graph data G, then this constraint will make the prior probability $P((u,v) \in E \mid A(v) = \text{'female'})$ high. At the same time, if the opposite is true for connections with male users, $P((u,v) \in E \mid A(v) = \text{'male'})$ will be low. From the perspective of node u, $P(G \mid A)$ is therefore biased.

Thus, DeBayes finds a debiased embedding as the Maximum Likelihood estimate $\mathbf{Z}_{deb} \triangleq \arg\max_{\mathbf{Z}} P(\hat{G} \mid \mathbf{Z}, \hat{A})$, where

$$P(\hat{G} \mid \mathbf{Z}, \hat{A}) = \frac{P(\mathbf{Z} \mid \hat{G})P(\hat{G} \mid \hat{A})}{P(\mathbf{Z} \mid \hat{A})}.$$

Note that DeBayes keeps the postulated distribution of the embedding conditional on the graph $P(\mathbf{Z} \mid \hat{G})$ the same as in CNE, i.e. independent of the attributes (thus $P(\mathbf{Z} \mid \hat{G}) = P(\mathbf{Z} \mid \hat{G}, \hat{A})$). This ensures that any bias is maximally absorbed by the prior. Taking Figure 1 as an example, if a political blog usually follows other blogs that are liberal-leaning, then it does not have to be embedded close to every other liberal blog. After all, it is was already known in the prior $P(G \mid A)$. Conversely, the few connections of the blog to conservative-leaning blogs will be modelled strongly in the embedding space.

4.3. Evaluating with an Oblivious Prior

After fitting the embeddings with a biased prior during training, the embeddings will contain far less information about the sensitive attributes. However, the posterior edge probabilities specified by $P(G \mid \mathbf{Z}_{deb}, \hat{A})$ are still computed using the biased distribution $P(G \mid \hat{A})$, and thus biased. To obtain debiased link predictions, the biased prior $P(G \mid \hat{A})$ can however simply be replaced with an *oblivious* prior $P(G)$ that is oblivious to the attribute $\hat{A}$. Providing node degrees

are independent of the sensitive attribute,[3] a suitable candidate is a prior with constraints (1). As this is a maximum entropy prior, sensitive attribute independence is guaranteed as long as the constraints are sensitive attribute independent. Thus we can directly use $\mathbf{Z}_{deb}$ to calculate the debiased link predictions using

$$P(G \mid \mathbf{Z}_{deb}) = \frac{P(\mathbf{Z}_{deb} \mid G)P(G)}{P(\mathbf{Z}_{deb})}.$$

To summarize, in DeBayes, $P(\hat{G} \mid \mathbf{Z}, \hat{A})$ is maximized during *training* w.r.t. $\mathbf{Z}$ to yield $\mathbf{Z}_{deb}$, after which $P(G \mid \mathbf{Z}_{deb})$ computed based on an oblivious prior can be used for debiased *evaluation* (link prediction).

4.4. Generalization

In Section 3.1, the assumption was made that every node has exactly one attribute value $A(v) = s$ with $s \in S$. However, it is a desirable property in fair network embedding that nodes can have multiple sensitive attributes that should all be protected from discrimination (Bose & Hamilton, 2019). In terms of notation, each user then has a vector of attribute values $[s_0, s_1, ... \mid s_0 \in S_0, s_1 \in S_1, ...]$, where each element has its own set of possible values.

DeBayes can easily be extended to such a case by adding the constraints from Eq. (2) for every S_i. Even with overlapping constraints, it is still possible to find a maximum entropy distribution. However, one should take care that the set of constraints do not fully specify G, as this will make it difficult for $\mathbf{Z}_{deb}$ to encode any useful information.

There is also the important case of graphs where not every attribute value is defined for every node. Such a case often occurs in recommender systems, where only the *user* nodes have sensitive attributes, while the *item* nodes do not. It then suffices to define an additional degree constraint for $P(G \mid A)$ that sums over those unattributed nodes.

5. Experiments

We ran our evaluation pipeline for 10 runs, with different random seeds and train/test splits. The training set always contained approximately 80% of the edges, with the test set containing the remaining 20% and an equal amount of non-edges. These test sets were then used to calculate the reported prediction AUC and high-level fairness scores.

Evaluation was done on two datasets, detailed in Section 5.1. A comparison was also made against two state-of-the-art baselines from the literature on fair network embedding as described in Section 2. Their specific configuration in these

Table 1. Dataset statistics. For every set of possible attribute values S their cardinality is given by $|S|$. The dataset names link to the URL where they can be downloaded.

DATASETS	$|V|$	$|E|$	S	$|S|$
DBLP	3,980	6,965	continent	5
ML-100K	2,625	100,000	gender	2
			age bracket	7
			occupation	21

experiments is described in Section 5.2. The fundamental concern of our experiments was to evaluate the performance of DeBayes on the level of network embedding and link prediction. These analyses are performed in Section 5.3 and Section 5.4 respectively. Finally, Section 5.5 wraps up with a discussion of the compared methods.

5.1. Datasets

There are two important use cases from link prediction that have been referred to throughout the paper: social graphs and recommender systems. The role of the former was filled by the *DBLP* dataset, a co-authorship network from the DBLP database. For recommender systems, we chose the popular *Movielens-100k* dataset, a network of user-movie ratings. Both datasets are summarized in Table 1.

DBLP The DBLP co-authorship network (Tang et al., 2008) is constructed from DBLP, a computer science bibliography database. In our analysis, we only considered publications at *KDD*, *ECML-PKDD*, *DAMI*, *NIPS*, *JMLR*, *ICML*, *MLJ* and *ICLR*. From the authors of papers accepted at these venues, we chose those for which the country of their affiliation could successfully be parsed from the 'org' field. The corresponding continents[4] are their sensitive attribute. Due to strong under-representation in the data we did not consider the continents of Africa and Antartica in our analysis. The authors were linked if they had collaborated at least once. Disconnected authors were removed.

Movielens-100k The Movielens-100k (ML-100k) dataset is a staple in recommender systems research due to its manageable size and rich data. The nodes represent users or movies and only user-movie links exist in the network. Since user-user links or movie-movie links never exist in this dataset, they were also not included in the test set. Furthermore, we ignored the 5-star ratings in our experiments. Of the user attributes, all are seen as sensitive: gender, age and occupation. Ages were binned into 7 age groups in order to make the attribute values categorical.

---

[3]Of course, this is not always the case. In such situations a density prior, only constraining the overall network density, could be used instead.

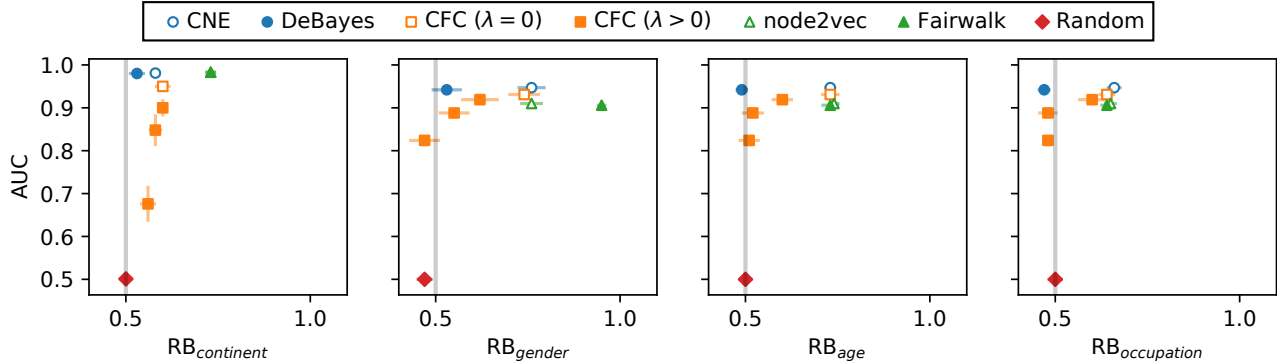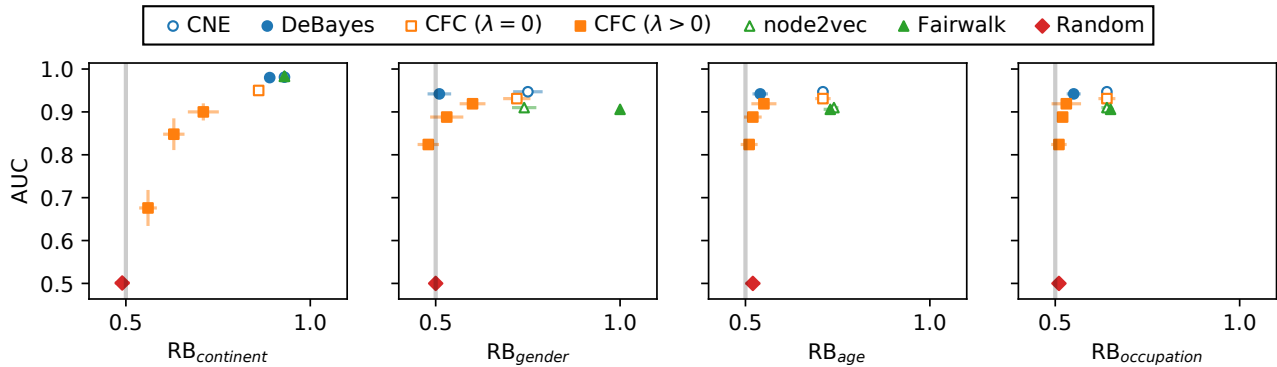[4]The country/continent mapping as defined by the `pycountry-convert` package.

(a) The RB when using a *logistic regression* classifier.

(b) The RB when using an *SVM* classifier with *RBF* kernel.

Figure 3. For various configurations of every method, the trade-off between link prediction AUC and RB when measured with classifiers of different strengths. Method configurations without any fairness adaptations, e.g. no biased training prior for DeBayes, were given **empty markers**. The ideal value for AUC is 1 and the ideal value for RB is 0.5 (indicated by the grey vertical line). The plotted values are the mean test set values over 10 random seeds. Symmetric error bars show standard deviation in both directions. Note that $RB_{continent}$ was evaluated on the DBLP dataset, while the other RB values result from experiments on ML-100k.

5.2. Configurations

Our proposed method is DeBayes. As baselines, *Compositional Fairness Constraints* (CFC) (Bose & Hamilton, 2019), *Fairwalk* (Rahman et al., 2019) and a uniformly *Random* predictor were used. Hyperparameter optimization was performed on the most influential parameters with link prediction AUC as a target. For that optimization, 20% of the training set was used as a validation set. Other parameters were left at their default values.

DeBayes The DeBayes method was trained using both an oblivious training prior and a biased training prior. Evaluation was always done using an oblivious training prior. Recall that DeBayes with an oblivious training and evaluation prior is equivalent to CNE. For dimensionality, $d = 8$ was chosen out of $d \in \{8, 16\}$. The parameter σ_2 was kept constant, and $\sigma_1 = 0.7$ was chosen from $[0.4, 0.9]$.

Compositional Fairness Constraints (CFC) We used the dot-product encoder for CFC in both datasets. The strength

of the adversarial term in the loss function is specified by the parameter λ. We trained models with $\lambda \in [0, 5, 25, 100]$. The maximum value was $\lambda = 100$, as that already resulted in a much lower prediction performance than the other algorithms. The learning rate parameter was evaluated over the range $[0.0001, 0.01]$. For DBLP, a learning rate of 0.01 was used; for ML-100k, it was left at its default value of 0.001.

Fairwalk Since Fairwalk can not be trivially extended to multiple sensitive attributes, it was only given 'gender' as a sensitive attribute for the ML-100k dataset. We present results for Fairwalk with and without the fair random walk modification. Note, the latter is equivalent to the original node2vec. The default dimensionality of $d = 128$ was used.

5.3. Analysis of Representation Bias

The representation bias (RB) (Section 3.2), is measured in our experiments using two classifiers: a logistic regression classifier and an SVM with a non-linear kernel. Their hyperparameters were tuned separately on every algorithm's set

of node embeddings and for every attribute. After tuning, each classifier is trained on 80% of the nodes and then tested on the remaining embeddings.

In Figure 3, the trade-off is shown between link prediction AUC and RB. By using a biased prior, DeBayes clearly produces significantly debiased embeddings with respect to all the ML-100k attributes (gender, age and occupation), since both classifiers can only achieve random-like performance in predicting them. At the same time, the decrease in AUC is limited. For DBLP, the logistic regression classifier becomes less successful at estimating the continent, but the non-linear SVM can still make a strong separation.

The embeddings from CFC's adversarial approach also appear to be well-debiased, even for the seemingly difficult case of reducing $RB_{continent}$ against a non-linear SVM classifier. However, a significant decrease in link prediction AUC is incurred at the same time. Furthermore, it was observed that on RB_{gender}, the bias in the embedding surprisingly *increased* using the fair random walk. In contrast to the other algorithms, both its AUC and RB_{gender} score are worse for the configuration with fair random walks. Recall that on ML-100k, only the 'gender' attribute was marked as sensitive for Fairwalk, explaining why measures for 'age' and 'occupation' remained relatively constant.

5.4. Analysis of High-Level Fairness

Results on the high-level measures are illustrated in Figure 4. On DBLP, the oblivious prior configuration for DeBayes already scores the best on the traditional fairness measures: DP and EO. Yet, when trained with a biased prior, the predictions appear even more fair. On the other measures, DeBayes also improves upon its unfair configuration.

Although CFC is also successful at improving on all fairness measures given a large enough λ value, it must do so with a sharply dropping AUC. Furthermore, Fairwalk maintains its strong AUC on both datasets, but with only a significant improvement on DP_{gender}. On the other hand, Fairwalk scores worse on EO_{gender} and ARP_{gender}.

Interestingly, while both DeBayes and CFC were able to reduce RB_{gender} to random-like values, they are unable to come close to the random predictor's high-level fairness. However, the achieved fairness measures were already quite low without any fairness adaptations.

5.5. Discussion

Both DeBayes and CFC are able to significantly debias the node representations, though they depend on different strategies. In the case of DeBayes, the learning process is allowed to freely minimize its loss function. As long as the prior is powerful enough to model the bias in the data well, even a non-linear classifier is unable to predict the sensitive
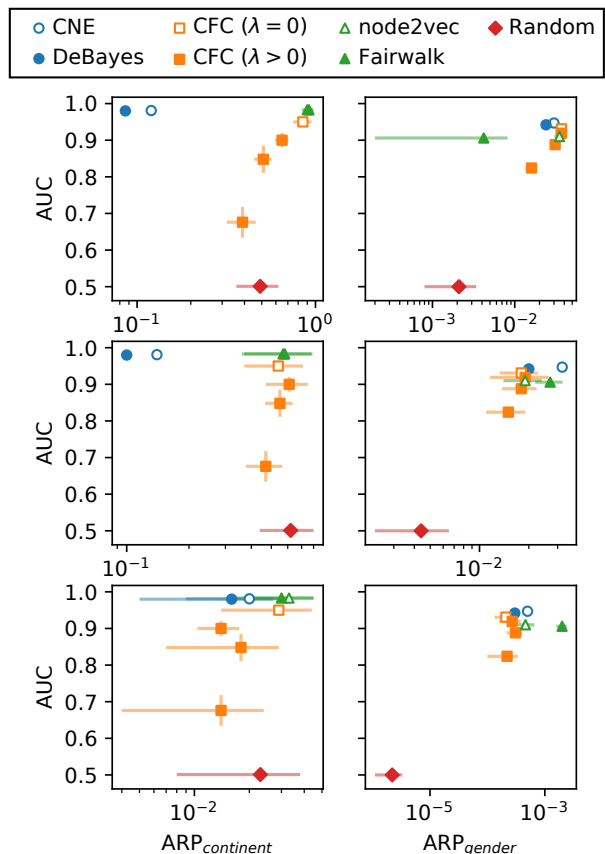

Figure 4. For various configurations of every method, the trade-off between link prediction AUC and high-level fairness measures. For ML-100k, the measures were only computed for the 'gender' attribute. In contrast to Figure 3, the ideal value for all high-level measures is 0. Also, each plot has a different, logarithmic x-axis.

information. The DeBayes method also generates more fair predictions by using an oblivious evaluation prior on top of its debiased embeddings. Overall, the results indicate that DeBayes indeed focusses on taking out biased information, while still learning informative embeddings.

This is in contrast with CFC, where an adversarial term causes the embeddings and downstream tasks to be more fair. The issue is that such a term could be unstable, complicating the learning of non-discriminatory dependencies between the ground truth and the data; a hypothesis that is supported by a sharp drop in AUC scores for comparatively mild fairness improvements. On the other hand, Fairwalk appears to make its embeddings more biased towards the sensitive attribute on ML-100k. It was only successful at lowering its DP, though it barely saw its AUC affected. Moreover, the fair random walk strategy did not have a significant impact on the DBLP dataset. One possible cause is the sparsity of the network, causing there to be many neighbourhoods where nodes are only connected with other nodes that have

the same attribute value. In such neighbourhoods, Fairwalk does not change the random walk compared to node2vec.

In addition to prediction performance and fairness, there are other useful properties for an algorithmic fairness method, such as *scalability*. Recall that for DeBayes, the prior distribution is computed completely before learning the embeddings. The latter step is equivalent to the original CNE method, which was shown to be faster than state-of-the-art network embeddings like node2vec on a single process/thread for large datasets (Kang et al., 2019). Furthermore, for the optimization problem of finding the prior distribution, DeBayes only adds constraints that are at most as complex as those in (1). They only increase the computational complexity by a constant factor and do not introduce additional memory requirements. However, we have so far only considered sensitive attributes with categorical values. If we had not performed binning of age groups for ML-100k, then the number of constraints (one for every age) would have become intractable. In order to take more complex sensitive attributes into account, such as those with continuous values, different constraints would have to be defined.

6. Conclusion

DeBayes is a conceptually simple and elegant method for finding debiased network embeddings based on Conditional Network Embedding. DeBayes results in substantially de-biased node embeddings in almost all cases, and the link predictions score well on three important high-level fairness measures. Even though there is inevitably a trade-off between fairness and prediction AUC, the cost in prediction quality by using DeBayes is limited. The method is also flexible enough to incorporate multiple types of sensitive attributes and does not depend on minimizing a fairness-related loss term. Instead, it depends on the ease of modelling the sensitive information through a prior distribution.

Further research into the training prior could allow for more kinds of sensitive attributes, e.g. with continuous values. Moreover, the choice of the evaluation prior is flexible and could be modified to be even more debiased (e.g. when dependencies exist between degrees and the sensitive attribute). Finally, principles underlying DeBayes could be generalized and applied to other areas in fair machine learning.

Acknowledgements

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.

Barocas, S. and Selbst, A. D. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

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

Calders, T. and Verwer, S. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pp. 3992–4001, 2017.

Creager, E., Madras, D., Jacobsen, J.-H., Weis, M., Swersky, K., Pitassi, T., and Zemel, R. Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning*, pp. 1436–1445, 2019.

Cui, P., Wang, X., Pei, J., and Zhu, W. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):833–852, 2018.

Ekstrand, M. D., Tian, M., Kazi, M. R. I., Mehrpouyan, H., and Kluver, D. Exploring author gender in book rating and recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 242–250, 2018.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 259–268, 2015.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., and Roth, D. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 329–338. ACM, 2019.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864. ACM, 2016.

Hamilton, W. L., Ying, R., and Leskovec, J. Representation learning on graphs: Methods and applications. *Bulletin of The IEEE Computer Society Technical Committee on Data Engineering*, 40, 2017.

Hardt, M., Price, E., Srebro, N., et al. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pp. 3315–3323, 2016.

Kamishima, T., Akaho, S., Asoh, H., and Sakuma, J. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 35–50. Springer, 2012.

Kang, B., Lijffijt, J., and De Bie, T. Conditional network embeddings. In *International Conference on Learning Representations*, 2019.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. Learning adversarially fair and transferable representations. In *International Conference on Machine Learning*, pp. 3384–3393, 2018.

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.

Palowitch, J. and Perozzi, B. Monet: Debiasing graph embeddings via the metadata-orthogonal training unit. *arXiv preprint arXiv:1909.11793*, 2019.

Pedreshi, D., Ruggieri, S., and Turini, F. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 560–568. ACM, 2008.

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.

Stoica, A.-A., Riederer, C., and Chaintreau, A. Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity. In *Proceedings of the 2018 World Wide Web Conference*, pp. 923–932. International World Wide Web Conferences Steering Committee, 2018.

Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., and Su", Z. "arnetminer: Extraction and mining of academic social networks". In *"KDD'08"*, pp. "990–998", 2008.

Woodworth, B., Gunasekar, S., Ohannessian, M. I., and Srebro, N. Learning non-discriminatory predictors. In *Proceedings of the 2017 Conference on Learning Theory*, Proceedings of Machine Learning Research. PMLR, 2017.

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, 2017.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *International Conference on Machine Learning*, pp. 325–333, 2013.