Noname manuscript No.
(will be inserted by the editor)

Feature-Blind Fairness in Collaborative Filtering Recommender Systems

Rodrigo Borges · Kostas Stefanidis

Received: 17 Aug 2020 / Revised: 09 Aug 2021 / Accepted: 19 Nov 2021

Abstract Recommender systems were originally proposed for helping users dealing with excessive amounts of data, by suggesting potentially interesting items to each one of them with the unique objective of achieving accurate predictions. These systems have diversified, have expanded to several domains, and were identified as generating biased results that could potentially harm the data items being recommended. The exposure in generated rankings, for instance in a job candidate selection situation, is supposed to be fairly distributed among candidates, regardless of sensitive attributes (gender, race, nationality, age) for providing equal opportunities. It can happen, however, that no such sensitive information is available in the data applied for training the recommender, and in this case, it is still possible to detect biases that can lead to unfair treatment, named *Feature-Blind* unfairness. In this work, we adopt Variational Autoencoders (VAE), considered as the state-of-the-art technique for Collaborative Filtering (CF) recommendations, and present a framework for addressing fairness when having only access to information about user-item interactions. More specifically, we are interested in Position and Popularity Bias. VAE loss function combines two terms associated to accuracy and quality of representation. We introduce a new term for encouraging fairness, and demonstrate the effect of promoting fair results despite of a tolerable decreasing in recommendations quality. In our best scenario, Position bias is reduced by 42% despite a reduction of 26% of recall in the top 100 recommendation results, when compared to the same situation without any fairness constraints.

Keywords Popularity Bias · Position Bias · Variational Autoencoders · Fairness · Fair Collaborative Filtering

Rodrigo Borges, Kostas Stefanidis
Tampere University, Finland
E-mail: {rodrigo.borges, konstantinos.stefanidis} @tuni.fi

1 Introduction

The amount of digital data produced in the Web increases each day, followed by the number of possibilities one has available when deciding to watch a movie, to hire a new employee or even to choose a romantic partner. It might be reasonable to say that when accessing an online platform, one has so many options available before making a decision that asking help from an intelligent system would turn necessary. Recommender systems were proposed in this context, for analyzing historical behavior and providing users with a subset of data items corresponding to their personal preferences.

In its most popular formulation, known as Collaborative Filtering (CF), recommender systems associate users to consumption profiles and close profiles are interpreted as similarity of preferences. The CF method is capable of inferring probabilities for each user/item pair based on neighboring users, assuming that similar users will behave similarly in the future. Finally, individual lists of suggestions are built with potentially interesting data items, which users have not seen yet. The aim of these systems used to be the prediction of potentially interesting data items with the highest accuracy possible, in order to satisfy and engage users. But the moment recommenders popularize and start being incorporated in many online systems, they need to account also for how fair their results are from the perspective of the data items being recommended.

In general, when systems are responsible for providing ranked lists the concept of fairness considers the superiority of higher positions in which data items are presented [20]. The position of an item is usually associated to how much attention it will get from users: the first positions concentrate much of the attention, and the attention decreases as the position gets higher. This is applicable, for instance, in the case of a search engines to which users submit queries, and get ordered list as a result. A fair result, in this case, is associated with having data items in the first positions of the rankings independently of the attributes considered sensitive. In recommenders, specifically, the historical behavior is stored and analyzed for producing individual lists of suggestions when requested by users. The results are personalized and can vary from one user to another, as well as from one round of recommendation to the next one.

Still, in recommenders, it can happen that the system will calculate relevance solely based on users interaction information, or it can also happen that sensitive information is not stored in databases due to privacy issues. Even in these situations, there is space for unfair recommendations, more specifically, through two types of bias, namely *Position Bias* and *Popularity Bias*. In Figure 1, for instance, we see examples of both biases extracted from a real dataset. On the left, the scores given to the top-10 items suggested to a random user are compared with a theoretical curve describing user's attention. From left to right, we see the level of attention decreasing as the position gets higher, whereas the scores remain practically stable. That is, equally good items are presented in substantially different levels of exposure (Position Bias). On the right, we calculate the number of times each movie was watched in the MovieLens dataset, and order them from the most to least popular, where we see

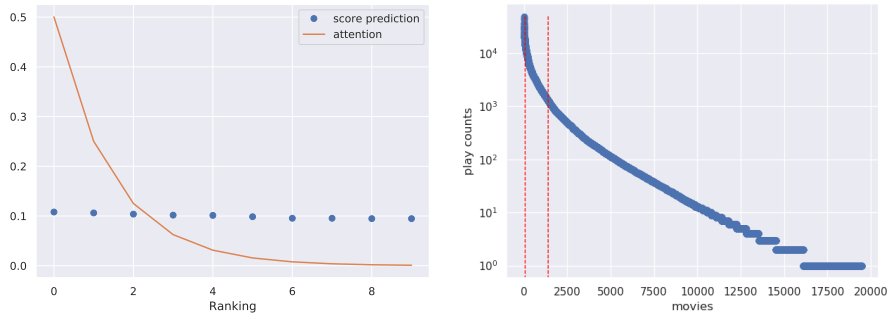

Fig. 1 Position (left) and popularity (right) bias in recommender systems. Left, a theoretical geometric attention decaying curve is compared to a real one in the case of the top-10 positions in a ranking for a random user in Movielens. Right, items are sorted by the number of interactions, and the vertical lines indicate 20% and 80% thresholds, indicating a very unbalanced distribution of popularity for the same dataset.

that few movies concentrate the majority of interactions from users. From left to right, the red lines indicate the thresholds for 20% and 80% of the whole distribution, respectively (Popularity Bias).

That said, we refer ourselves to methods for mitigating biases with no connection to any sensitive information, as promoting *Feature-Blind fairness*. Specifically, we refer to situations where biases are observed independent of any individual feature of users or items, and we solely exploit statistical information of interaction between them, this way, having different implications than hiding sensitive information during the training phase [14]. So, the challenge here is to train a model for personalized recommendations that is able to provide users with potentially interesting data items, while ensuring that all items are being exposed to users as fair as possible. Encouraging the recommender to predict scores proportionally to the attention items will receive from users, should maintain popular items in the highest positions in the ranking, preserving the overall accuracy of the system. At the same time, it should attract less popular items to the subsequent positions, promoting items that would normally be located in positions of lower exposure.

Variational Autoencoders (VAE) are considered today the state-of-the-art for CF recommender systems, due to its accuracy, scalability and robustness for dealing with extremely big datasets. Users' features are learned from the data, in what is known as the encoder phase, before propagating through the decoder where scores are actually attributed to each item. VAE recommenders are derived from the inference model and are capable of learning its latent variables according to a loss function which combines accuracy and quality of representation. When encouraged to learn the first, it is basically adjusting its predictions to the ground truth data, and when encouraged to learn the second it is approximating a theoretical assumed distribution to the one observed in the data. Both terms can have different weights during the training

allowing the autoencoder to prioritize one of the objectives, providing it with flexibility regarding its parameters. In this work, we propose adding a new term responsible for achieving fair results, that will act in the same fashion, encouraging the weights to be learned according to a formula previously defined. The amount of bias to be removed in the learning process, here referred as a proxy for unfairness, can also be controlled through an extra parameter tested in the experiments.

Specifically, we propose a framework for mitigating bias, or unfairness, in recommendations, where features about the items are not known, and therefore, cannot be judged as being discriminatory. We refer to this scenario as *Feature-Blind*, and we argue that when this happens there is still space for biased results, according to principles inherited from individual fairness, where equally relevant individuals are treated similarly. Our approach is more of a statistical approach dedicated to ensuring fair treatment of items according to the consumption information contained in the dataset.

In short, our main contributions in this work are:

- We introduce the problem of ensuring fairness in recommendations when no information about data items is available, named *Feature-Blind*, with special attention on position bias and popularity bias.
- We present a framework that allows configuration of how much bias is to be removed from the recommendation results for the price of reduced accuracy.
- We experimentally show that the strategy proposed here is capable of reducing position bias from the results in a higher proportion than the accuracy decreases. And in some cases the same trade-off is observed for reducing also popularity bias.

The remaining of this paper is organized as follows. In Section 2, we present a review on recent works about fairness in decision making algorithms, and we focus on the specific ones related to recommender systems. In Section 3, we formalize the task of providing users with fair recommendation results from the perspective of data items, and we define Feature-Blind fairness. In Section 4, Position Bias and Popularity Bias are presented in details, as examples of unfairness that might emerge from suggestions when no sensitive information of data items is provided. In Section 5, we present a framework for mitigating unfairness in recommenders implemented with VAE. Experiments with real datasets are described in Section 6, as well as competitors methods and metrics for evaluating the results. Finally, the results are presented and discussed in Section 7, and conclusions and future work are presented in Section 8.

2 Previous Work

The urge on adapting decision making algorithms to become explicitly fairness-aware becomes first evident in situations of automatic classification, before expanding to ranking and recommender systems. In this section, we review

the main concepts related to fair classification, ranking and recommendation algorithms. We present a discussion on techniques that hide demographic information from the intelligent systems to analyze the effect in reducing the bias and discrimination. We also summarize a recent discussion about generating representations considered fair with the technique considered the state-of-the-art approach for Collaborative Filtering, named Variational Autoencoders.

Fairness in Classification. When it comes to the discussion of algorithmic fairness, the first setup addressed was the supervised classifiers. In a simplified scenario, classifiers are assumed as generating binary outputs, say positive and negative, and a fair outcome was first proposed as a trade-off between individual and group fairness [10]. *Individual fairness* assume similar individuals being represented equally along the positive results. *Group fairness*, also referred as statistical parity, assume individuals separated in groups according to a sensitive attribute (e.g., gender, age, race), and these groups being selected equally by the classifier. It has been showed that both can be compatible if groups are homogeneous, or can demand a prioritization in the case groups are different one from the other. Having the final decision independent of the protected attribute is also referred as *demographic parity*.

A different formulation for fair classification differentiates *equality of odds* and *equality of opportunity* [11]. Both concepts will arise from the argumentation that demographic parity can fail to promote fair results when (i) one of the groups being classified is too small, or (ii) when the results correlate with the sensitive attribute. Equality of odds will allow the predicted score to depend on these attributes, but only through the target label. In other words, it encourages the use of features allowing the prediction of the output, but prohibits abusing the sensitive variable as a proxy for the prediction. Equality of opportunity will extend the previous definition and require non-discriminatory attitude only within the advantaged group.

All concepts of fairness presented so far operate under the premise that information about sensitive features are known in advance, and can be used to detect biased outcomes. But it can happen that protected class membership is not observed in the data, for legal, operational or behavioral reasons [14]. Some institutions may not allow ethnicity information to be collected in registration forms, bad quality forms can also conflict with self identification of race and gender, and it can also happen that people are reluctant to inform their race for fear of potential discrimination.

Under these circumstances it might be necessary to fill the missing data with a proxy model, which is capable of guessing the class information based on a specific or a set of features. It was demonstrated that this can also lead to biased outcomes [7], as a result of a complex interaction of multiple different biases contained in the data.

We, instead, propose measuring unfairness in a situation when no demographic information is available, not even in a secondary dataset or proxy model. When this happens, there are no protected groups to take into consideration as a prior, and the unfairness is calculated exclusively through statistics

obtained from the data, representation and results, according to what we are here referring to as a *Feature-blind* approach.

Fairness in Rankings. Moving to the domain of ranking solutions, the output is not considered anymore as a selection to a category but as a list of items in order of relevance, to be selected by the user of the algorithm. The problem is modeled as in a search engine, where one can submit a query and get a list with the results ordered by relevance. This time, there are no good or bad outcomes as in the binary classification, but rather better or worse positions in the ranking. We can still apply all previous concepts of fairness if we consider, for example, a ranking process as a first step of a classification process, within which the best subset of items is selected. But even then, new challenges are imposed to practitioners, different from the ones mentioned so far. Next, we focus on describing specific situations for evaluating fairness in ranked lists.

The ratio of protected individuals that appear within a prefix of the ranking must be above a given proportion, in order to satisfy statistical tests of representativeness [26]. One possible solution to this is to re-rank items after the scores were calculated in order to balance opportunity. Furthermore, the attention received by the items in different positions in the ranking is not the same: items ranked in first positions are exposed to much more attention than the lower ones [8]. The situation of having homogeneous scores given in the first positions in a ranking is mentioned as promoting position bias in [5]. This is described as the situation of originating unfairness due to the wide difference between attention (position) and relevance (score): the difference of attention changes drastically from the first position to the second, but the same difference is not observed between the relevance values. Approximating both distributions through a post processing method is described as promoting *equity of attention*.

The idea of distributing users' attention along items in a fair manner is adopted in our work, and transposed to the domain of recommendation. Furthermore, we extend its applicability by suggesting that, when introduced as a loss function in a recommendation process, it can mitigate also another source of unfairness, the popularity bias.

Fairness in Recommendations. As mentioned before, recommender systems have some specificities when compared to general purpose ranking systems as in the case of, for example, search engines. When a user submits a query to a search engine it explicitly represents the information needed, whereas in a recommendation scenario the task is to provide users with items they might like, based on implicit information collected previously [3]. The collaborative approach, specific for recommenders, is also prone to bias already in its first assumption: grouping similar users together will most likely approximate frequent users, and isolate them from sparse ones. From the perspective of items, popular items will also influence the training processes if error is measured by classical accuracy, due to popularity bias [1, 2, 29].

When it comes to the techniques applied in the prediction processes, recommenders also demand new approaches for measuring and removing bias, and consequently promoting fairness. The popular matrix factorization is pointed as potentially unfair due to popularity bias [1,21,3]. New metrics for measuring fairness in recommendation are presented in [25], different from the ones proposed for ranking systems. The idea of compensating an unfair recommendation round with the following ones is explored in [22], in the context of group recommendations.

There are several metrics available for measuring unfairness in recommendation, we selected some of the most popular ones, and we pinpoint the reasons why they can not be considered in our study. *MADr (Mean Average Difference - rating)* [28] is the absolute difference between mean ratings of different groups, assuming two groups. But in our case, we do not separate items or users in groups. Instead, we measure the exposure or popularity associated with each item. *BS (Dataset Bias)*, *BR (Recommendation Bias)* and *BD (Bias Disparity)* [23] refer to categories of items and protected users groups. In our work, we do not have categories of items or demographics about users. As a consequence there is no such notion of groups. *MADR (Mean Average Difference - Ranking)* and *GCE (General Cross-Entropy)* [9] are also measures applicable to situations where a sensitive attribute is defined, and recommendation results are evaluated according to it, and they are not applicable in our context.

We also review some of the metrics that are useful for measuring popularity in the context of recommendation systems, and we report the reasons why they cannot be applied here. *APLT (Average Percentage of Long Tail items)* and *ACLT (Average Coverage of Long Tail items)* [2] measure the average percentage of long tail items in the recommended lists as a proxy of coverage or diversity. ACLT measures what fraction of the long-tail items the recommender has covered. But in our work, there is no separation of the popularity distribution in regions, and consequently no long tail. Instead, we use a continuous measure considering the popularity of each item separately. *RSP (Ranking-based Statistical Parity)* [29] can be defined as forcing the ranking probability distributions of different item groups to be the same, and *REO (Ranking-based Equal Opportunity)* encourages the true positive rates (TPR) of different groups to be the same. In our case, again, we do not have items separated in groups.

In our work, we explore a mechanism for removing biases from recommendation results, or more precisely we add a new term to the loss function of VAE, considered today as the state-of-the-art for CF recommenders [17].

Fair Representations. In some cases, fairness is considered a matter of representation, and techniques for isolating sensitive attributes in the main task being addressed. A central idea in these approaches is to define an attribute in the input data that should be neutralized, and adapt the loss function applied in the learning process so the intermediate representation of the data satisfies a constraint associated to fairness.

One first attempt was proposed in [27] when a loss function was adapted for generating fair classification results. A similar approach is described in [19] but applying VAE to learn the representations. A tensor decomposition technique is proposed by [28], able to isolate sensitive features and provide recommendation results uncorrelated to them. Recently, β-VAE became popular due to its capacity of disentangling the latent variables learned from the input data [18,12]. The main idea is to enhance the power of inner representation in the sense of increasing mutual information during the learning process. These representations are demonstrated as providing fair results due to its capacity of isolating potential distributions used for generating the input data, in a explainability fashion [18]. [6] adds a stochastic component to the regular operation of VAE in order to mitigate position bias in a CF task. The hypothesis is tested when applying three different Gaussian noise distributions for achieving different levels of fluctuations in the final recommendation rankings.

Our proposed model is able to learn fair representations from users' behavior data while retaining as much information about the input as possible. This is done in a similar fashion than [19], but applied for recommendations and with no information about sensitive attributes.

3 Feature-Blind Fairness

Let us assume a group of users ($u \in U$) interacting with items ($n \in N$), and every interaction user/item stored in a rating matrix ($X \in \mathbb{N}^{|U|\times|N|}$). A recommender is trained having X as the input, and after having its weights optimized it assigns probability values to each unseen item/user pair. A subset ($K \in \mathbb{N}$) of the best options is presented to each user in a descending order according to the predicted score. The score assigned by the algorithm reflects the item relevance ($r \in [0,1]$), and the position in the ranking is used as a proxy for attention ($a \in [0,1]$) in a way that lower positions are exposed to more attention than higher ones. That is, the most likely items occupy the first positions, and the score decreases as the position index increases ($a^l_{ip} > a^l_{iq}$ as well as $r^l_{ip} > r^l_{iq}, \forall n_{ip}, n_{iq}$ with $p < q$). This implies the first positions in the ranking as the most relevant, and also as the ones more exposed to users attention. All variables are defined in Table 1.

Table 1 Variables definition

$n \in N$	a set of items to be ranked
$u \in U$	a set of users
$a \in [0,1]$	the position in the ranking (a proxy for the level of attention)
A_j	the attention distribution associated to a single list presented to user u_j
$r \in [0,1]$	the score given by the model (a proxy for the relevance)
R_j	the relevance distribution associated to a single list presented to user u_j
$r^j_{ik} \in [0,1]$	relevance score of item n_{ik} in ranking for user u_j in position k
$a^j_{ik} \in [0,1]$	attention to which item n_{ik} is exposed in ranking for user u_j in position k

In the core, we require that ranked subjects receive attention that is proportional to their relevance in a series of rankings. The requirement is presented in [5] as *Equity of Attention*, and is defined as:

$$\frac{\sum_{l=1}^{U} a_{ip}^{l}}{\sum_{l=1}^{U} r_{ip}^{l}} = \frac{\sum_{l=1}^{U} a_{iq}^{l}}{\sum_{l=1}^{U} r_{iq}^{l}}, \forall n_{ip}, n_{iq} \text{ with } p < q \tag{1}$$

For example, the relation between the attention to which the item in the first position is exposed to (a_{i1}^{l}) and its relevance (r_{i1}^{l}) should be as similar as possible to the relation measured for the item in the second position. And it should be also valid for all other items in the set. Achieving equal proportions is not a feasible option, as we will see in the following sections, but the difference should be minimized accordingly.

In this work, we are interested in the specific situation when no demographic information is available for modeling, as in the case of many recommenders training processes. We refer to the unfairness that might originate from biased results as a situation of *Feature-Blind* unfairness. This kind of unfairness can be detected as a direct consequence of biases originated in situations where users are interacting with items, or from any premise on the predicted scores, as for example in a search engine or a recommender system.

Definition 1: Feature-Blind criteria are the ones applied for measuring unfairness when no demographic information is taken into account.

It can happen that ranking relevance is calculated taking into account the clickthrough rate an item received previously. It can also happen that an item that is very popular in a recommendation scheme, and is constantly appearing in the first positions. In both cases, marginal items are, in principle, excluded from the privileged ranking positions, and as long as the bias increases, the chance of overcoming it becomes harder each time. It can also happen that new items are added to the platform, and present no previous information about user interaction. The lack of a strategy to attract these items to public attention will prevent them being exposed properly.

Difference from Fairness Under Unawareness. These criteria differ from *Fairness Under Unawareness* [14] in the sense that there is no proxy model here, and items and users are never associated to their inner characteristics. We are specially interested in statistical biases that can flourish from the methods, from the metrics applied for measuring utility or from the interaction between users and items.

Proximity to Individual Fairness. The proposed feature-blind criteria refer to each item individually, even though the bias can have been measured according to some statistical value extracted from the whole population. There is still space for decreasing, for example, demographic parity in a method optimized for reducing a feature-blind criterion.

4 Position and Popularity Bias

We now describe two biases, position and popularity, considered as potential sources of unfairness in recommendations. We bring practical examples when both can occur, and formulas for measuring them in the final results.

4.1 Position Bias

When providing recommendations to users, algorithms are responsible for assigning probability values to each item in the set, and presenting them in a descending order. It can happen, however, that one individual ranking presents very homogeneous regions, that is, items with very similar relevance occupying different positions. This situation is referred to as *Position Bias* [5], and when occurring systematically and for many rankings, can promote long term unfairness. Next, we describe a situation in which it can potentially occur.

Situation 1: Lets assume that a recommendation algorithm was previously trained and is ready to provide suggestions of movies. An specific user opens the recommendation interface and sees a list of 5 movies sorted from the most to the least relevant. The user does not have access to this information, but the scores given to each of the movies, in order, were 0.9, 0.9, 0.89, 0.8, 0.79. We can imagine this happening for several times and for several users, a situation where equivalently relevant items (with same or really close scores) being exposed to considerably different levels of attention in a recommendation ranking.

That said, we state that: *A recommender is fair as long as equally relevant items are presented to users in a corresponding position in the ranking. In other terms, as long as it can provide rankings with relevance proportional to the attention received by users.*

The position in the ranking is assumed as proxy for the attention, and the relevance as a proxy for the score given by the system. The attention is defined as a geometric distribution [5], the first position is assumes as concentrating majority of the attention, and attention value decreases according to a parameter p within the interval $[0, 1]$:

$$w_j = \begin{cases} p(1-p)^{j-1}, & \text{if } j \leq k \\ 0, & \text{if } j > k \end{cases} \tag{2}$$

Lets consider the k items predicted with highest scores by the recommendation algorithm as having relevance values $[r_1, r_2, \ldots, r_k]$, or R, and corresponding attention levels $[a_1, a_2, \ldots, a_k]$ or A, calculated with Formula 2. A and R are converted to multinomial probability distributions by simply dividing each term by the summation of all values ($A/sum(A)$ and $R/sum(R)$), as in the example of Figure 1 (Left). The divergence between both is calculated with Kullback-Leibler (KL) divergence formula:

$$POSB@K(u_j) = D_{KL}(A_j||R_j) = \sum_{k=1}^{K} P(a_k^j) \log \left(\frac{P(a_k^j)}{P(r_k^j)} \right) \quad [1] \tag{3}$$

KL divergence has its origin in the field of information theory, using the idea of entropy. It measures the expectation of the log difference between the probability of data in the original distribution with the approximating distribution. Here, A is selected as the target distribution, and KL divergence will retrieve small values in the case R is similar to it.

The attention distribution is held fixed, with a static value for p, and for every ranking, and the value calculated by POSB@K indicate how close the scores calculated for the first K items distributions are to this theoretical attention distribution.

4.2 Popularity Bias

Usually, in Collaborative Filtering recommendation systems, few data items concentrate the majority of ratings given by users, referred to as *Popularity Bias*. And the consequences are that a great proportion of unpopular items, the ones with few ratings, end up sharing small percentages of users feedback. We assume the popularity bias as a mixture of unbalanced preferences authentically expressed by the users, as well as a side effect of algorithms and metrics applied by these systems. Moreover, suggesting unpopular items has the desired effect of serendipity (providing users with novelty), and also expand the knowledge of the system about unpopular items with very few rating information. We follow describing another situation in which it can potentially occur.

Situation 2: Let's assume a recommender operating through an algorithm trained according to an error-based metric, that is to say, its success refers to the ratio of right guesses it can perform in a separated part of the data (test set). After N rounds of recommendations, 0.7% of available items were responsible for 20% of users interactions registered by the platform (Figure 1). We expect that in its next train round, the algorithm will try to adjust its weights to maximize its overall accuracy, which will certainly refer mostly to those 0.7% items than for unpopular ones responsible, for 99.3% of the play counts. We imagine this happening successively, and in each round the model is more adjusted according to popular items, and unaware of a great slice of items that could potentially found their niches of consumption, or simply refer to items added to the platform recently.

Formally: *A recommneder is fair as long as it can attract unpopular items to users attention. In others terms, it can distribute users' attention as equally as possible among items.*

In order to measure Popularity Bias, we propose a metric inspired by NDCG that expresses how much a ranking is biased because of the popularity

[1] The i index for indicating item n_i in position i is removed for the sake of simplicity.

of recommended items. As a first step, a discounted summation of popularity is calculated for the top-k items with Discounted Cumulative Popularity (DCP):

$$DCP@K = \sum_{i=1}^{K} \frac{\omega(i)}{\log_2(i+1)} \tag{4}$$

where ω indicates a function for measuring the proportion of interactions in the training set associated to item in position i, and this number is considered as a proxy its popularity. High values of DCP indicate popular items being presented in first K positions.

The ideal version of DCP, named IDCP, is calculated with the formula just presented, but this time having the same set of items ordered by popularity. The popularity bias (POPB) is obtained as a normalized version of DCP when considering IDCP:

$$POPB@K = \frac{DCP}{IDCP} \tag{5}$$

5 Fair Recommendations

Variational Autoencoders (VAE) are considered today the state-of-the-art technique for CF recommendation solutions [17]. VAE derive directly from Auto Encoding Variational Bayes (AEVB) [15], which apply Stochastic Gradient Variational Bayes (SGVB), allowing efficient approximation of posterior inference and learning model parameters without the need of expensive iterate inference schemes per datapoint. Briefly said, Variational Bayes approximates the full posterior by attempting to minimize the Kullback-Leibler divergence between the true posterior and a predefined factorized distribution on the same variables, as described in the following.

5.1 Variational Autoencoder

Let the observed variable $\mathbf{x}$ be a random sample from a process whose true distribution $p(\mathbf{x})$ is unknown. Our aim is to approximate the process with a model $p_\theta(\mathbf{x})$ with parameters θ. $p_\theta(\mathbf{x})$ can be very complex (contain arbitrary dependencies), and a common approach is to assume an unobserved random latent variable $\mathbf{z}$ involved in the process of generating $\mathbf{x}$. A simple assumption is $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z}) = p_\theta(\mathbf{x})p_\theta(\mathbf{z}|\mathbf{x})$, where $p_\theta(\mathbf{x}|\mathbf{z})$ corresponds to estimating $\mathbf{x}$ from $\mathbf{z}$, and $p_\theta(\mathbf{z}|\mathbf{x})$ corresponds to estimating $\mathbf{z}$ from $\mathbf{x}$.

$p_\theta(\mathbf{z})$ is assumed as Gaussian, but $p_\theta(\mathbf{z}|\mathbf{x})$ is still intractable. An auxiliary model $q_\phi(\mathbf{z}|\mathbf{x})$ is introduced then, whose parameters ϕ will be learned to approximate $q_\phi(\mathbf{z}|\mathbf{x}) \sim p_\theta(\mathbf{z}|\mathbf{x})$, and (reproduced from [16]):

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}(\log p_\theta(\mathbf{x})) \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big(\log\Big(\frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\Big)\Big) \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big(\log\Big(\frac{p_\theta(\mathbf{x},\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\frac{q_\phi(\mathbf{z}|\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x})}\Big)\Big) \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big(\log\Big(\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\Big)\Big) + \mathbb{E}_{q_\theta(\mathbf{z}|\mathbf{x})}\Big(\log\Big(\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\Big)\Big)\end{aligned} \tag{6}$$

The second term in the right hand side is the non-negative Kullback-Leibler divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z}|\mathbf{x})$, and the first term is known as the *evidence lower bound* (ELBO). ELBO is defined as:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}(\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})) \tag{7}$$

And maximizing the ELBO corresponds to maximizing:

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \log p_\theta(\mathbf{x}) - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) \tag{8}$$

The first term in the right corresponds to the to the marginal likelihood, and the second term the error of distribution approximation. The optimization process corresponds to optimizing parameters ϕ and θ. In the specific case of a rating-matrix-based recommender, $\mathbf{x}_u$ contains the number of interactions for user u, $q_\phi(\mathbf{z}|\mathbf{x}_u)$ corresponds to the estimation of $\mathbf{z}$ space departing from input data, named Encoder, and $p_\theta(\mathbf{x}_u|\mathbf{z})$ corresponds to estimating the original data departing from the latent space, named Decoder.

5.2 Bias-aware VAEs

Our aim here it to introduce a new term to the ELBO for encouraging the optimization process to generate fair results, as a consequence of bias removal. We add a new term to Equation 8 referred as bias, to be minimized together with prediction error and KL divergence. We also add regularization factors for the second and third term, respectively β and λ.

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \log p_\theta(\mathbf{x}) - \beta \cdot \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})) - \lambda \cdot Bias \tag{9}$$

The framework implemented from this equation allows the user to define the bias to be removed form the results, and also the strength of the quality of representation (KL divergence) and fairness (Bias).

The bias measured as the distance between attention (A) and relevance (R) distributions, as in [5], but measured in a training batch of size U_T with $U_T << U$.

$$Bias = \sum_{i=1}^{N} \left\| \sum_{j=1}^{U_T} a_i^j - \sum_{j=1}^{U_T} r_i^j \right\| \tag{10}$$

Rankings are considered independent of each other, each one being associated to one user[2]. The resources needed for the calculation are the scores given by the algorithm and the position in the sorted ranking. High measured values indicate high relevance items in low positions, or low relevance in high positions in the rankings.

6 Experiments

In order to evaluate our methods, we run a series of recommendation experiments with data obtained from movie and music recommendation platforms. We choose two among the most popular datasets in recommendation field, MovieLens-20M[3] and Netflix [4], and two smaller ones associated to music consumption, Nowplaying[4] and 30music [24].

6.1 Data Preparation

Nowplaying dataset is a compilation of tracks that were posted on Twitter with the hashtag #nowplaying, 30music is also a compilation of play events collected from the LastFM platform, the MovieLens-20M dataset contains movie ratings collected from 1995 to 2015, and the Netflix data has a similar format and was collected from 1998 to 2005. Users who interacted with less than 5 song/movies were removed and the data was converted to binary as in the case of implicit feedback. In the following, we describe the process of preparing the data, adjusting parameters for the model and competitors, and the metrics used in the experiments.

We consider each user as a ranking round [3], and remove sequences longer than 1,000 items for avoiding unrealistic behaviors. The model selected for attention is a geometric progression with $p = 0.5$ (Equation 2), meaning that the first position takes the value of p and decreases exponentially towards 0.

For the Nowplaying dataset, there are 496,657 watching events from 12,621 users and 33,167 movies (sparsity: 0.119%). In 30music, there are 1,228,485 watching events from 25,038 users and 86,398 movies (sparsity: 0.057%). In the case of Movielens, there are 9,785,141 watching events from 136,526 users and 13,160 movies (sparsity: 0.545%). In the case of Netflix, there are 54,514,109 watching events from 459,559 users and 17,680 movies (sparsity: 0.671%). All information is presented in Table 2.

The set of users is split in train/validation/test subsets, in the following percentages 80/10/10. In the case of the training subset, all items consumed by each user u is considered as a profile P_u used for adjusting the models' weights. Validation and test subsets are also converted in profiles, but this time these profiles are also randomly split in query/target subsets, in the percentages

[2] This could be measured for a sequence of recommendations for a single user as well.

[3] https://grouplens.org/datasets/movielens/

[4] https://zenodo.org/record/2594483#.YDzqBHVfhhE

Table 2 Datasets description

Dataset	# events	# users	# items
NowPlaying	496,657	12,621	33,167
30 Music	1,228,485	25,038	86,398
MovieLens	9,785,141	136,526	13,160
Netflix	54,514,109	459,559	17,680

80/20. We refer to the subset of queries associated to test users as Q^{Te} and the targets associated to the same users as T^{Te}. The same procedure is applied for the users separated for validating the model.

6.2 Method and Baselines

The train is conducted with a batch size of 500 samples, the validation and test batches are set to 100 samples. Encoder and Decoder are implemented as one-hidden MLP, and the model is trained for 300 epochs in all cases. The MLP dimensions depend on the number of items available for the recommender (N), and is described as $[N \rightarrow 600 \rightarrow 200 \rightarrow 600 \rightarrow N]$. The learning rate is set to 0.001 and its value decreases by a factor of 0.1 consecutively in epochs number 100 and 150. We add a dropout of 0.5 as a first layer in the Encoder, and a Tanh between layers in both Encoder and Decoder.

[18, 12] are here considered as adopting a Feature-Blind criterion, and taken as baseline methods. Specifically, we adopt the approach proposed in [12]. These approaches were all applied in the context of image representation, and they propose a technique for isolating independent factors of variation in the input data in order to avoid biases for sensitive factors that one might not be even aware. The approach presented in [19], however, assumes an explicit sensitive attribute and can not be applied here. We do not apply annealing steps to out training processes, and in this case, the effect of disentanglement, as proposed in [12], is achieved by varying the value of β in Equation 9.

We train the model for three different values of β (0.1, 1.0 and 10) with no bias removal, and maintain the first one considered here a standard operation of VAE in the following experiments. The extra factor responsible for mitigating the bias is then incorporated to the training process with three different values for λ (25, 50, 100) for three more rounds of experiment. In every experiment the model is trained for 250 epochs and after each epoch the model is validated by presenting every entry in the validation queries subset (Q^{Vl}). A list of recommendations is obtained ordered by relevance and truncated in the 100th position. The output scores generated by the model are always normalizes with a softmax function for obtaining probabilities.

Matrix factorization methods were widely applied for the task of CF, and a specific adaptation was proposed for dealing with implicit feedbacks[5] [13],

[5] Implicit feedbacks are unintrusively acquired as part of the users' interaction process (i.e. click, watch, skip), as opposed to explicit feedback that require an active action of rating items.

adopted in this work as WMF. Hyperparameters were maintained with the values reported in the original work, except for the one associated to *confidence*, that was set as linear and with α equals to 100 through several rounds of experiments conducted with the smallest dataset.

6.3 Evaluation Metrics

The quality of recommendations is measured comparing the predicted scores with the target subset of test subset. The truncated version of Recall is adopted from [17] for indicating accuracy.

$$RECALL@K(u) = \frac{1}{min(K, |T_u^{Te}|)} \sum_{k=1}^{K} \mathbb{I}[n_k \in T_u^{Te}] \tag{11}$$

where $\mathbb{I}$ is an indicator function, n_k the item ranked in position k and T_u^{Te} is the target subset for user u. Recall indicate the proportion of items brought in the first K position that were actually in the target subset, and does not consider the order in which items are shown.

ARP (Average Recommendation Popularity) os proposed in [2] for measuring the average popularity of the recommended items in each list. We adapted the original formulation to a normalized version.

$$ARP@K(u) = \frac{1}{I} \sum_{k=1}^{K} \omega(i) \tag{12}$$

where ω indicates a function for measuring the number of interactions in the training set associated to item in position i, as in Equation 5, and I represents the total number of interactions (number of user-item interactions) observed in the training set. This metric will help on the understanding of the proportion of interactions concentrated in the first K positions of recommended lists having the total interactions in the train subset as a reference.

In order to explicitly measure the overall effect of reducing accuracy while removing bias from the results, we measure two trade-offs, the $POSB-TF$ and the $POPB-TF$. The first one is an average between relative reduction of position bias and decrease in recall, when comparing to the unbiased setup as a reference:

$$\text{POSB-TF} = \left(\frac{POSB@100_{GT}}{POSB@100} + \frac{REC@100}{REC@100_{GT}} \right) \times \frac{1}{2} \tag{13}$$

where the subscript GT stands for *Ground Truth* and is here assumed as the situation when β equals to 0.1. Higher values indicate a positive effect of reducing bias in a higher proportion than recall reduction. In the second case, with $POP-TF$, the same logic is applied to popularity bias, with the following formula:

$$\text{POPB-TF} = \left(\frac{POPB@100_{GT}}{POPB@100} + \frac{REC@100}{REC@100_{GT}} \right) \times \frac{1}{2} \tag{14}$$

The same notation remains from the former case, and again, values higher than 1 indicate the superiority of bias removal despite of a accuracy decrease. The Recall was elected as a reference for reflecting directly the accuracy of the method [21].

7 Results

We start by analyzing the curves indicating the evolution of Recall, Position Bias, Popularity Bias and Average Recommendation Popularity during the training phase when β was set to 0.1 and λ was set to 0, 25, 50 and 100. Figure 2 shows the impact of increasing the amount of bias correction in the system's accuracy. The best overall accuracy results were obtained for the Movielens dataset, and the lowest one was observed in the case of Nowplaying. In this first three datasets, the RECALL@100 measurements stabilize around 150 epochs and, in the case of Movielens, a sudden slope is observed in epoch 100, due to the reduction in the learning rate by a factor of 0.1. In the case of Netflix, the accuracy measurements diverge for high values of λ, and in the worst scenario, not even a stability is achieved.

Reducing bias from the recommendation results has an interesting effect in the measurements of POS@100, as one can see in Figure 2. The very first thing to be mentioned is the clear effect of bias removal in the results measured for the validation set. It decreases as the value of λ increases, as expected, but in a different scales: when training the recommenders with music datasets the POSB is not affected when increasing λ from 0 to 25, as much as when increasing it from 25 to 50, or from 50 to 100. There is also a clear difference regarding stability when the four datasets' partial results are compared. In the three first ones, the measurements get stable after a certain amount of epochs, and in the last, it starts increasing after epoch 150. This might indicate different sensibility to learning rate reduction in this specific epoch or in this dataset.

Similar results are observed when tracking the removal of Popularity Bias, except that now the measurements remains relatively stable after decreasing the learning rate. In the case of movie consumption datasets, the ARP values seems to mimic POPB ones, which is understandable, once they are both associated with the the same phenomena. But when training the models with music data, both seem to behave a bit more independent from each other.

Finally, we apply the trained models to the test subset, in order to check the generalization capacities of those models. We report results in Table 3 according to the metrics presented in Sections 4 and 6. We now have two more competitors, WMF and POP. The best overall accuracy results were obtained for β equals to 0.1 in the case of movie datasets, and higher values for the parameter provided better results in the remaining experiments with music

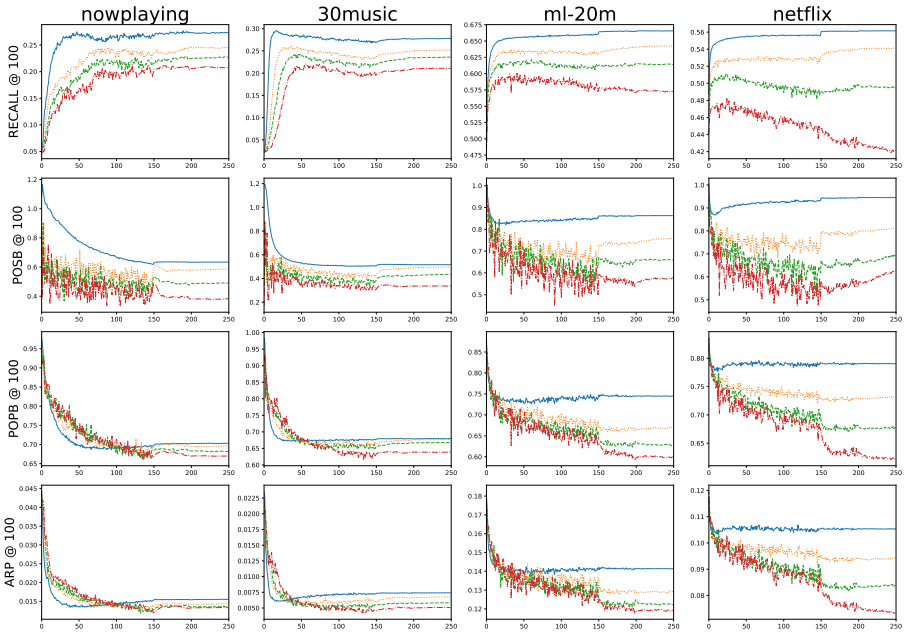

Fig. 2 All results measured during the training processes for Nowplaying, 30music, MovieLens and Netflix datasets. The curves correspond to λ equals to 0 (blue/solid), 25 (orange/dotted), 50 (green/dashed) and 100 (red/dash-dotted). It is worth mentioning that λ equals to 0 correspond to standard VAE with no bias regularization.

listening data. Accuracy, represented here by Recall, decreases relatively fast when increasing the same parameter, as one can also notice in the trade-off values. The metrics for measuring biases, however, increase in the same fashion for all datasets.

Higher values for λ are responsible for better POS-TF trade-off in the case of Nowplaying and 30music datasets. The trade-off for popularity bias, on the other hand, decreases in a similar trend. The best overall RECALL@100 results are observed for WMF in both cases, and in the case of 30music it is reflected also in the best POS-TF value. But the best trade-off was measured for Nowplaying trained with λ equals to 100, when RECALL@100 was reduced by 26% and POSB@100 by 42%.

In the case of the models trained with MovieLens data, the best trade-off between decreasing the Recall while decreasing the position bias is achieved for λ equals to 100, when the RECALL@100 decreases 14% for promoting a reduction of 33% in the POS@100. In the case of the bias originated from unbalanced popularity of items, the POP@100 was reduced by 22%.

The situation is different in the case of Netflix dataset, when the best trade-off for position bias was also observed for the same value of λ, 100, but the best scenario when reducing the bias associated to popularity was observed for λ equals to 50. In the first case RECALL@100 had its value decreased by 24%,

Table 3 Results for performance and bias metrics. VAE($\beta = 0.1$) is reported as VAE.

Dataset	Method	REC@100	POSB@100	POPB@100	ARP@100	POSB-TF	POPB-TF
NowPlaying	POP	0.055	31.542	1.000	0.045	0.116	0.472
	WMF	0.330	1.206	0.746	0.010	0.908	**1.123**
	VAE	0.261	0.667	0.733	0.016	-	-
	VAE($\beta = 1.0$)	0.268	0.760	0.759	0.018	0.952	0.997
	VAE($\beta = 10.0$)	0.269	0.766	0.765	0.018	0.951	0.995
	VAE($\lambda = 25$)	0.233	0.597	0.725	0.014	1.004	0.951
	VAE($\lambda = 50$)	0.222	0.484	0.710	0.013	1.113	0.941
	VAE($\lambda = 100$)	0.194	0.388	0.702	0.013	**1.232**	0.894
30Music	POP	0.024	18.756	1.000	0.025	0.058	0.401
	WMF	0.432	1.203	0.739	0.005	0.991	**1.250**
	VAE	0.282	0.542	0.715	0.007	-	-
	VAE($\beta = 1.0$)	0.287	0.614	0.715	0.008	0.951	1.009
	VAE($\beta = 10.0$)	0.284	0.626	0.714	0.008	0.937	1.005
	VAE($\lambda = 25$)	0.251	0.526	0.707	0.007	0.961	0.952
	VAE($\lambda = 50$)	0.231	0.429	0.691	0.006	1.042	0.928
	VAE($\lambda = 100$)	0.211	0.340	0.666	0.005	**1.171**	0.910
MovieLens	POP	0.332	inf	1.000	0.219	0.250	0.649
	WMF	0.556	1.137	0.764	0.094	0.792	0.941
	VAE	0.664	0.850	0.799	0.124	-	-
	VAE($\beta = 1.0$)	0.645	0.935	0.835	0.134	0.941	0.965
	VAE($\beta = 10$)	0.638	0.956	0.850	0.138	0.925	0.951
	VAE($\lambda = 25$)	0.643	0.824	0.727	0.114	1.000	1.034
	VAE($\lambda = 50$)	0.611	0.684	0.663	0.108	1.082	1.063
	VAE($\lambda = 100$)	0.568	0.567	0.624	0.104	**1.178**	**1.068**
Netflix	POP	0.273	inf	1.000	0.166	0.244	0.664
	WMF	0.407	1.259	0.783	0.078	0.734	0.900
	VAE	0.560	0.932	0.840	0.099	-	-
	VAE($\beta = 1.0$)	0.532	1.031	0.867	0.109	0.927	0.960
	VAE($\beta = 10$)	0.520	1.055	0.880	0.113	0.906	0.942
	VAE($\lambda = 25$)	0.539	0.848	0.782	0.089	1.031	1.019
	VAE($\lambda = 50$)	0.495	0.690	0.713	0.079	1.117	**1.031**
	VAE($\lambda = 100$)	0.425	0.599	0.653	0.069	**1.157**	1.023

and POS@100 is reduced by 36%. In the second case the accuracy decreases by 12%, and POP@100 by 15%.

In order to bring the reader a visualization of the literal effect of removing bias from CB recommendations, we select a random user and show the first 10 predicted scores before and after applying the new term responsible for bias removal. The comparison can be seen in Figures 3 and 4. When increasing the new term in Equation 9 for encouraging the system to remove the bias from the results, the model is actually approximating its predictions to a theoretical attention curve (Equation 2). The result is clear in Figure 3. The same interpretation is also valid for mitigating Popularity Bias, but this time the effect is of attracting unpopular items to the first positions of the ranking, as one can see in Figure 4.

8 Conclusions

In this work, we revisited several definitions of fairness proposed in different fields of research, for considering the situation when no demographic (and no sensitive) information about users is provided in the data. We refer ourselves to this situation as a common one in the recommendation field, when datasets are restricted to users and items interactions, and when there is still space for biases and unfair results. We then proposed new criteria for the so called

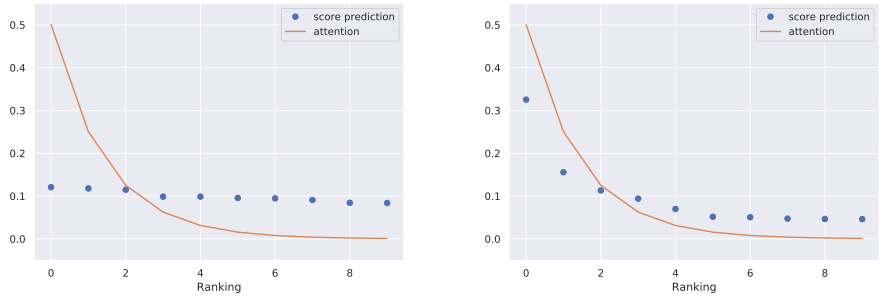

Fig. 3 The top 10 scores calculated for a random user before (Left) and after (Right) applying the new term for removing Position Bias. The attention (solid line) is calculated by a theoretical model, and the predictions are plotted as dots.

Fig. 4 The top 10 scores calculated for a random user before (Left) and after (Right) applying the new term for removing Popularity Bias. The popularity (solid line) is calculated by summing the ratings an item received in the training subset, and the predictions are plotted as dots.

Feature-Blind fairness, and we discuss possible relations with previous definitions. We analyzed the trade-offs between accuracy and fairness in Collaborative Filtering recommendations. We introduced a framework within which the designer is capable of tuning parameters depending on how much bias is to be removed, and how much accuracy should be maintained. The method is based on Variational Autoencoders, which provides the basis for generating high quality recommendations.

An interesting effect was observed when reducing the learning rate by a factor of 0.1 in the epoch number 150: the Position Bias, here calculated as POS@100, started increasing after a strong decrease trend. The effect was observed when applying positives results for the parameter responsible for removing bias from the results (λ), and the rate the bias increases gets higher for greater values of λ. In the case of MovieLens, the effect was also observed for values of λ greater than 100, but not presented in the text. These led us to consider the fact that different datasets have different sensibility to the reduction of the learning rate, and that higher values of λ might require longer training processes, or at least different intervals for reducing the learning rate.

As a final remark, the bias term proposed in this work correlates directly the bias presented as being associated to position in the recommendation ranking, but has proven also efficient in removing popularity bias.

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Controlling popularity bias in learning-to-rank recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017, pp. 42–46. ACM (2017). DOI 10.1145/3109859.3109912. URL https://doi.org/10.1145/3109859.3109912
2. Abdollahpouri, H., Burke, R., Mobasher, B.: Managing popularity bias in recommender systems with personalized re-ranking. In: Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019, pp. 413–418. AAAI Press (2019). URL https://aaai.org/ocs/index.php/FLAIRS/FLAIRS19/paper/view/18199
3. Bellogín, A., Castells, P., Cantador, I.: Statistical biases in information retrieval metrics for recommender systems. Inf. Retr. J. **20**(6), 606–634 (2017). DOI 10.1007/s10791-017-9312-z. URL https://doi.org/10.1007/s10791-017-9312-z
4. Bennett, J., Lanning, S., Netflix, N.: The netflix prize. In: In KDD Cup and Workshop in conjunction with KDD (2007)
5. Biega, A.J., Gummadi, K.P., Weikum, G.: Equity of attention: Amortizing individual fairness in rankings. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, pp. 405–414. ACM (2018). DOI 10.1145/3209978.3210063. URL https://doi.org/10.1145/3209978.3210063
6. Borges, R., Stefanidis, K.: Enhancing long term fairness in recommendations with variational autoencoders. In: R. Chbeir, Y. Manolopoulos, S. Ilarri, A. Papadopoulos (eds.) 11th International Conference on Management of Digital EcoSystems, MEDES 2019, Limassol, Cyprus, November, 2019, pp. 95–102. ACM (2019). DOI 10.1145/3297662.3365798. URL https://doi.org/10.1145/3297662.3365798
7. Chen, J., Kallus, N., Mao, X., Svacha, G., Udell, M.: Fairness under unawareness: Assessing disparity when protected class is unobserved. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019, pp. 339–348. ACM (2019). DOI 10.1145/3287560.3287594. URL https://doi.org/10.1145/3287560.3287594
8. Craswell, N., Zoeter, O., Taylor, M.J., Ramsey, B.: An experimental comparison of click position-bias models. In: M. Najork, A.Z. Broder, S. Chakrabarti (eds.) Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008, pp. 87–94. ACM (2008). DOI 10.1145/1341531.1341545. URL https://doi.org/10.1145/1341531.1341545
9. Deldjoo, Y., Anelli, V.W., Zamani, H., Bellogin, A., Di Noia, T.: A flexible framework for evaluating user and item fairness in recommender systems. User Modeling and User-Adapted Interaction (UMUAI) (2021). Https://doi.org/10.1007/s11257-020-09285-1
10. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.S.: Fairness through awareness. In: S. Goldwasser (ed.) Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012, pp. 214–226. ACM (2012). DOI 10.1145/2090236.2090255. URL https://doi.org/10.1145/2090236.2090255
11. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: D.D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, R. Garnett (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pp. 3315–3323 (2016). URL http://papers.nips.cc/paper/6374-equality-of-opportunity-in-supervised-learning
12. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings (2017)

13. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08, pp. 263–272 (2008)
14. Kallus, N., Mao, X., Zhou, A.: Assessing algorithmic fairness with unobserved protected class using data combination. In: M. Hildebrandt, C. Castillo, E. Celis, S. Ruggieri, L. Taylor, G. Zanfir-Fortuna (eds.) FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, p. 110. ACM (2020). DOI 10.1145/3351095.3373154. URL https://doi.org/10.1145/3351095.3373154
15. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Y. Bengio, Y. LeCun (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings (2014)
16. Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Foundations and Trends® in Machine Learning **12**(4), 307–392 (2019). DOI 10.1561/2200000056
17. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: P. Champin, F.L. Gandon, M. Lalmas, P.G. Ipeirotis (eds.) Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, pp. 689–698. ACM (2018). DOI 10.1145/3178876.3186150. URL https://doi.org/10.1145/3178876.3186150
18. Locatello, F., Abbati, G., Rainforth, T., Bauer, S., Schölkopf, B., Bachem, O.: On the fairness of disentangled representations. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 14584–14597 (2019). URL http://papers.nips.cc/paper/9603-on-the-fairness-of-disentangled-representations
19. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.S.: The variational fair autoencoder. In: Y. Bengio, Y. LeCun (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings (2016)
20. Pitoura, E., Stefanidis, K., Koutrika, G.: Fairness in rankings and recommendations: An overview. The VLDB Journal (2021)
21. Steck, H.: Item popularity and recommendation accuracy. In: B. Mobasher, R.D. Burke, D. Jannach, G. Adomavicius (eds.) Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011, pp. 125–132. ACM (2011). DOI 10.1145/2043932.2043957. URL https://doi.org/10.1145/2043932.2043957
22. Stratigi, M., Nummenmaa, J., Pitoura, E., Stefanidis, K.: Fair sequential group recommendations. In: SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, online event, [Brno, Czech Republic], March 30 - April 3, 2020, pp. 1443–1452. ACM (2020). DOI 10.1145/3341105.3375766. URL https://doi.org/10.1145/3341105.3375766
23. Tsintzou, V., Pitoura, E., Tsaparas, P.: Bias disparity in recommendation systems. In: Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019, vol. 2440 (2019). URL http://ceur-ws.org/Vol-2440/short4.pdf
24. Turrin, R., Quadrana, M., Condorelli, A., Pagano, R., Cremonesi, P.: 30music listening and playlists dataset. In: RecSys Posters (2015)
25. Yao, S., Huang, B.: Beyond parity: Fairness objectives for collaborative filtering. In: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp. 2921–2930 (2017). URL http://papers.nips.cc/paper/6885-beyond-parity-fairness-objectives-for-collaborative-filtering
26. Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., Baeza-Yates, R.: Fa*ir: A fair top-k ranking algorithm. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017, pp. 1569–1578. ACM (2017). DOI 10.1145/3132847.3132938. URL https://doi.org/10.1145/3132847.3132938
27. Zemel, R.S., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: Proceedings of the 30th International Conference on Ma-

chine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, *JMLR Workshop and Conference Proceedings*, vol. 28, pp. 325–333. JMLR.org (2013). URL http://proceedings.mlr.press/v28/zemel13.html

28. Zhu, Z., Hu, X., Caverlee, J.: Fairness-aware tensor-based recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, pp. 1153–1162 (2018). DOI 10.1145/3269206.3271795. URL https://doi.org/10.1145/3269206.3271795
29. Zhu, Z., Wang, J., Caverlee, J.: Measuring and mitigating item under-recommendation bias in personalized ranking systems. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, p. 449–458. Association for Computing Machinery, New York, NY, USA (2020). DOI 10.1145/3397271.3401177. URL https://doi.org/10.1145/3397271.3401177