



Feature-blind fairness in collaborative filtering recommender systems

Rodrigo Borges¹ · Kostas Stefanidis¹

Received: 17 August 2020 / Revised: 12 January 2022 / Accepted: 15 January 2022 /
Published online: 22 February 2022
© The Author(s) 2022

Abstract

Recommender systems were originally proposed for suggesting potentially relevant items to users, with the unique objective of providing accurate suggestions. These recommenders started being adopted in several domains, and were identified as generating biased results that could 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 their sensitive attributes (gender, race, nationality, age) for promoting 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, there is still space for 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 we 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 with 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 decrease in recommendation quality. In our best scenario, position bias is reduced by 42% despite a reduction of 26% in recall in the top 100 recommendation results, compared to the same situation without any fairness constraints.

Keywords Popularity bias · Position bias · Variational autoencoders · Fairness · Fair collaborative filtering

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

✉ Rodrigo Borges
rodrigo.borges@tuni.fi

Kostas Stefanidis
konstantinos.stefanidis@tuni.fi

¹ Tampere University, Tampere, Finland

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 turn out 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 with consumption profiles, and similar profiles are interpreted as similarity of preferences. The CF method is capable of inferring probabilities for each unseen 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, according to predicted values. 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 into many online systems, they need to be designed considering also 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 with how much attention it will attract from users: the first positions concentrate much of the attention, and the attention level 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 Fig. 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

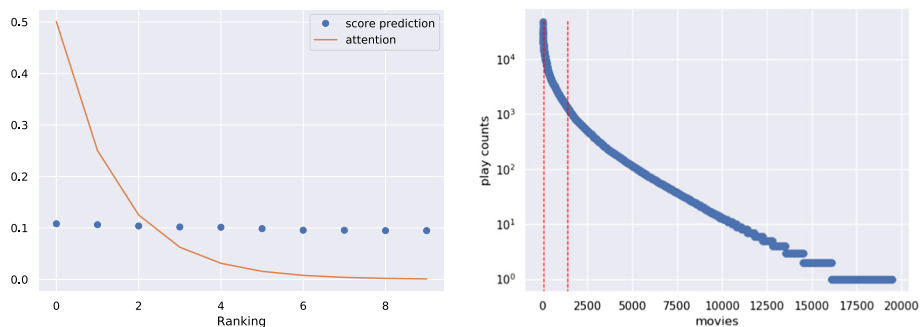


Fig. 1 Position (left) and popularity (right) bias in recommender systems. Left, a theoretical geometric attention decaying curve is compared to a real one in the case of the top 10 positions in a ranking for a random user in MovieLens. Right, items are sorted by the number of interactions, and the vertical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

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 with substantially different levels of exposure (Position Bias). On the right, we calculated the number of times each movie was watched in the MovieLens dataset, and we ordered them from the most to the least popular. Here, we see 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 association 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 exclusively exploit statistical information of interaction between them, thus having different implications than hiding sensitive information during the training phase [14]. 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. Here, the recommender is encouraged to predict scores that are proportional to the attention level associated with its position in the suggested list. This way popular items are expected to appear in the first positions of the list, whereas less popular items are expected to occupy the subsequent positions, with reasonable levels of exposure.

Variational Autoencoders (VAE) are considered today the state-of-the-art approach for CF recommender systems, due to its accuracy, scalability and robustness for dealing with extremely big datasets. In VAE's regular operation, first low-dimensional user features are learned from user/item interaction data (encoder), and then, these features are propagated back to their original dimensions (decoder). VAE recommenders are derived from an inference model and are capable of learning its latent variables (user features) according to a loss function that combines accuracy and quality of representation. When encouraged to learn the first term, VAE's internal weights are adjusted to the ground truth data, and when encouraged to learn the second term, the weights are adjusted for approximating its latent variables to Gaussian distributions. Both terms can have different weights during the training procedure, allowing the autoencoder to prioritize one of the two objectives. In this work, we propose the addition of a new term to the VAE loss function, responsible for approximating the generated scores to a theoretical attention curve. This new term is responsible for mitigating the bias (assumed here as unfairness) in the results. The amount of bias to be removed during the learning process can also be controlled through an extra parameter.

We propose a framework for mitigating bias, or unfairness, in recommendation scenarios where features about items are not known, and therefore, cannot be applied for measuring discrimination. We refer to this scenario as *Feature-Blind*, and we argue that when that is the case, there is still space for biased results according to principles inherited from individual fairness [10]. Our approach is more of a statistical approach dedicated to ensuring fair treatment of items according to the consumption information available for the recommendation system.

In short, our main contributions in this work are:

- We introduce the problem of ensuring fairness in recommendation scenarios when no information about data items is available, named *Feature-Blind*. We pay special attention to position bias and popularity bias.
- We present a framework that allows the 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 in 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 Sect. 2, we present a review on recent works about fairness in decision making algorithms, and we focus on the ones related to recommender systems. In Sect. 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 Sect. 4, Position Bias and Popularity Bias are presented in details, as examples of unfairness that might emerge in situations where no sensitive information about data items is provided. In Sect. 5, we present a framework for mitigating unfairness in recommenders implemented with VAE. Experiments with real datasets are described in Sect. 6, as well as competitors methods and metrics for evaluating the results. Finally, the results are presented and discussed in Sect. 7, and conclusions and future work are presented in Sect. 8.

2 Previous work

The urge on adapting decision making algorithms to become explicitly fairness-aware is first evident in situations of automatic classification, and only after that new methods are proposed for fair 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 designed for hiding demographic information from intelligent systems, with the aim of reducing the bias and discrimination. We also summarize a recent discussion on generating user representations considered fair, which can be applied to any subsequent task, ensuring that biases were reduced or neutralized.

Fairness in Classification. Looking back to the discussion of algorithmic fairness, the first setup addressed was the one of 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 among 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 heterogeneous. Having the final decision independent of the protected attribute is also referred to as *demographic parity*.

A different formulation for fair classification differentiates *equality of odds* and *equality of opportunity* [11]. Both concepts emerge from the argument 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 allows the predicted score to depend on these attributes, but only through the target label. In other words, it encourages the use of features in the task of predicting the output, but prohibits abusing the sensitive variable as a proxy for the predictions. Equality of opportunity extends the previous definition and requires 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 identify 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 because of 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 between 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 algorithms, the output is not considered as a binary decision anymore, the results are now presented as a list of items ranked by relevance. This time, there are no good or bad outcomes, as in the case of category prediction, but rather better or worse positions in a ranked list. We can still apply all previous concepts of fairness if we consider, for example, a ranking procedure 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 [5]. The situation might be considered unfair 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 user attention among 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, e.g., the popularity bias.

Fairness in Recommendations. Recommender systems have some specificities when compared to general purpose ranking systems, as in the case of, for example, search engines. When users submit queries to search engines they are explicitly expressing 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 was pointed out as potentially unfair due to popularity bias [1, 3, 21]. 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 cannot be considered in our study. *MADr (Mean Average Difference-rating)* [28] is the absolute difference between mean ratings of different groups, assuming two groups of users. 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. These are also 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. User/item interaction data is sometimes considered in short-head and long-tail regions, for expressing popularity, and *APLT (Average Percentage of Long Tail items)* and *ACLT (Average Coverage of Long Tail items)* [2] measure the 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 the state-of-the-art approach for CF recommenders [17].

Fair Representations. In some cases, fairness is considered a matter of representation, and techniques are proposed for isolating sensitive attributes while calculating user or item features. 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 in a way that the intermediate representation of the data satisfies a constraint associated with fairness.

One first attempt was proposed in [27], where a loss function is adapted for generating fair classification results. A similar approach is described in [19] but applying VAE to learn these representations. A tensor decomposition technique is proposed in [28] that is able to isolate sensitive features and provide recommendation results uncorrelated to them. Recently, β -VAE became relatively popular due to its capacity of disentangling the latent variables learned from the input data [12, 18]. The main idea is to enhance the power of inner representation in the sense of increasing mutual information during the learning process. These representations were demonstrated as providing potentially fair results due to its capacity of identifying potential distributions used for generating the input data, in an 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.

The model proposed here is able to learn fair representations from user 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

We assume a group of users ($u \in U$) interacting with items ($n \in N$), and every user/item interaction being stored in a rating matrix ($X \in \mathbb{N}^{|U| \times |N|}$). A recommender is trained having X as input, and after having its weights optimized, it assigns probability values to each unseen item/user pair. A subset ($K \in \mathbb{N}$) containing the most likely items 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 ($a_{ip} > a_{iq}$ as well as $r_{ip} > r_{iq}, \forall n_{ip}, n_{iq}$ 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.

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}^1) and its relevance (r_{i1}^1) 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 might be observed 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.

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 attention level)
A_j	The attention distribution associated with a single list presented to user u_j
$r \in [0, 1]$	The score given by the model (a proxy for relevance)
R_j	The relevance distribution associated with a single list presented to user u_j
$r_{ik}^j \in [0, 1]$	Relevance attributes to item n_i in a ranking presented to user u_j , in position k
$a_{ik}^j \in [0, 1]$	Attention to which item n_i is exposed to in a ranking for user u_j , in position k

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 based on the clickthrough rate an item received previously. It can also happen that an item that is very popular in a recommendation scheme 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 chances of overcoming it becomes lower with time. It can also happen that new items are added to the platform, with no previous information about user interaction. The lack of a strategy to attract these items to public attention will prevent them to be 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 with 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 recommendation results.

4.1 Position bias

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

Situation 1: We assume a recommendation algorithm that was previously trained, and that is ready to provide suggestions of movies. A 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) are 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 generate 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],

where the first position is assumed as concentrating majority of the attention, and attention value decreases according to a parameter p within the interval $[0, 1]$:

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

The k items predicted with highest scores by the recommendation algorithm have relevance values denoted as $R = [r_1, r_2, \dots, r_k]$, and attention levels denoted as $A = [a_1, a_2, \dots, a_k]$, calculated with Formula 2. A and R are converted to multinomial probability distributions by simply dividing each term by the summation of all values ($A/\text{sum}(A)$ and $R/\text{sum}(R)$), as in the example of Fig. 1 (Left). The divergence between both is calculated with Kullback–Leibler (KL) divergence formula¹:

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

KL divergence has its origin in the field of information theory, and it measures the expectation of the log difference between the probabilities of an original distribution, and an approximated 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 the values calculated for POSB@K indicate how close the distribution generated for the top- k scores and a theoretical attention distribution are.

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 consequence is that a great proportion of unpopular items, the ones with few ratings, end up associated with a small percentage of users feedback. We assume the popularity bias as a combination of unbalanced preferences, authentically expressed by the users, and 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. We now describe a situation in which popularity bias can potentially occur.

Situation 2: Assume a recommender operating through an algorithm trained according to an error-based metric, that is to say, a metric whose success is assumed as 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 (Fig. 1). We expect that in its next training round, the algorithm will try to adjust its weights to maximize its overall accuracy. It might be reasonable to imagine that most of these adjustments will concern those 0.7% items, despite of unpopular items responsible for 99.3% of play counts. We imagine this happening successively, and at each round, the model is being more and more adjusted according to popular items.

Formally: *A recommender 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.*

¹ The i index for indicating item n_i is removed for the sake of simplicity.

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 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 associated with item in position i in the training set, and this number is considered as a proxy for 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 same formula, but this time having the same set of items ordered by popularity. The popularity bias (POPB) is obtained as a normalized version of DCP, considering IDCP:

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

5 Fair recommendations

Variational Autoencoders (VAE) are considered today the state-of-the-art approach for CF recommendation [17]. VAEs derive from Autoencoding 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]):

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

The second term in the right-hand side is the nonnegative 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 to maximize, the ELBO corresponds to maximize:

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

In the right-hand side, the first term corresponds to the marginal likelihood, and the second term corresponds to the error of distribution approximation. Parameters ϕ and θ are optimized jointly. In the specific case of a rating-matrix-based recommender, \mathbf{x}_u denotes the interactions profile of user u , $q_\phi(\mathbf{z}|\mathbf{x}_u)$ corresponds to the estimation of \mathbf{z} space based on the input data (Encoder), and $p_\theta(\mathbf{x}_u|\mathbf{z})$ corresponds to the estimation of the original data based on the latent space (Decoder).

5.2 Bias-aware VAEs

We introduce a new term to the ELBO (Equation 8) for encouraging the optimization process to generate fair results, as a consequence of bias removal. The new term is minimized together with prediction error and KL divergence. We also add hyperparameters for controlling the strength of the second and third terms, respectively, β and λ .

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

Parameter β is associated with the weight attributed to the quality of representation, and parameter λ is associated with the weight attributed to the bias in the recommendation results.

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

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

The rankings presented to each user are considered independent from each other. The resources needed for calculating the bias are the scores given by the algorithm and the position in the sorted ranking.

6 Experiments

In order to evaluate our proposition, we run a series of recommendation experiments with data obtained from movie and music recommendation platforms. We selected two popular movie datasets, MovieLens-20M² and Netflix [4], and two smaller ones associated with music consumption, Nowplaying³ and 30music [24].

² <https://grouplens.org/datasets/movielens/>.

³ <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

6.1 Data preparation

Nowplaying dataset is a compilation of tracks that were posted on Twitter with the hashtag #nowplaying, and 30music is also a compilation of play events collected from the LastFM platform. MovieLens-20M dataset contains movie ratings collected from 1995 to 2015, and the Netflix dataset contains also movie rating, collected between 1998 and 2005. Users who interacted with less than 5 song/movies were removed, and the data were 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 considered each user as a ranking round [3], and we removed 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 assumes the value set for p , and attention decreases exponentially toward 0.

In Nowplaying dataset, there are 496,657 listening events associated with 12,621 users and 33,167 tracks (sparsity: 0.119%). In 30music, there are 1,228,485 listening events associated with 25,038 users and 86,398 tracks (sparsity: 0.057%). In the case of MovieLens, there are 9,785,141 watching events associated with 136,526 users and 13,160 movies (sparsity: 0.545%). And in the case of Netflix, there are 54,514,109 watching events associated with 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 80/20. We refer to the subset of queries associated with test users as Q^{Te} and the targets associated with 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, and the validation and test batches are set to 100 samples. Encoder and decoder are implemented as one-hidden multilayer perceptron (MLP), and the model is trained for 250 epochs in all cases. The MLP dimensions depend on the number N of items available for the recommender, and it 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 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, as proposed in [17].

The methods proposed in [12, 18] are considered as adopting a Feature-Blind criterion, and they were elected as baseline methods. Specifically, we adopt the approach proposed in [12]. Both approaches were originally proposed in the context of image representation, and they

present techniques for isolating independent variation factors associated with the input data. A similar process is applied here for avoiding biases toward sensitive factors which one might not be even aware. The approach presented in [19], however, assumes an explicit sensitive attribute and cannot be applied here. The disentanglement effect, as proposed in [12], is achieved here by varying the value of β in Equation 9.

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

Matrix factorization methods were widely applied to the task of CF, and an adaptation was proposed for dealing with implicit feedbacks⁴ [13]. This specific adaptation is considered here as a baseline method, named WMF. Hyperparameters were maintained with the values reported in the original work, except for the one associated with *confidence* that was set as linear and with α equals to 100. These values were determined as the best ones after 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 separated for test. The truncated version of Recall was 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) was proposed in [2] for measuring the average popularity of the recommended items in the output 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 is a function for measuring the number of interaction associated with item in position i , as in Equation 5; and I represents the total number of user/item interactions; both observed in the training set. This metric will help us understand the proportion of interactions concentrated in the first K positions of recommended lists, having the total interactions in the training subset as a reference.

In order to 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 the average of a relative reduction in position bias and a decrease in recall:

⁴ 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.

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

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

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

The notation from the former case is maintained, and again, values higher than 1 indicate a superior bias removal despite of a decrease in accuracy.

7 Results

We start describing 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 strength of the bias term in the system’s accuracy. The best overall accuracy results were obtained for the MovieLens, and the lowest ones were observed in the case of Nowplaying 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

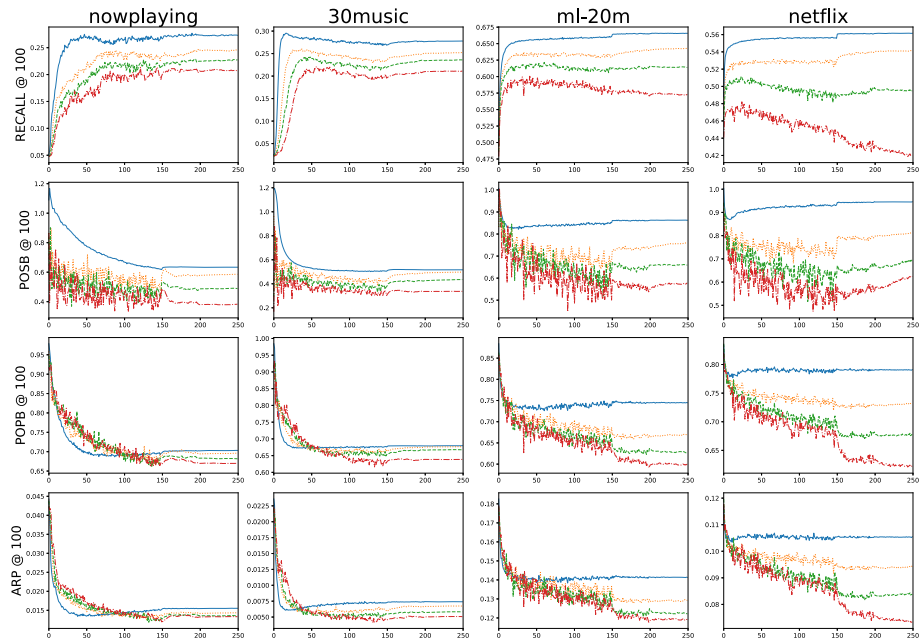


Fig. 2 All results measured during the training processes for Nowplaying, 30music, MovieLens and Netflix datasets. The curves correspond to λ equals to 0 (blue/solid), 25 (orange/dotted), 50 (green/dashed) and 100 (red/dash-dotted). It is worth mentioning that λ equals to 0 correspond to standard VAE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

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 shown in Fig. 2. The first thing to be mentioned is the clear effect of bias removal in the results measured for the validation set. Position biases decrease 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 well 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 one, it starts increasing after epoch 150. This might indicate different sensibility to learning rate reduction in this specific epoch or in one specific 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 seem to mimic POPB ones, which is understandable, once they are both associated with 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 capacity of those models. We report results in Table 3 according to the metrics presented in Sects. 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 attributed to this parameter provided better results in the remaining experiments with music 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-offs 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 when λ was set equals to 100, and trained with Nowplaying dataset. In this setup, RECALL@100 was reduced by 26% and POSB@100 by 42%.

In the case of the models trained with MovieLens data, the best POSB-TF trade-off is achieved for λ equals to 100, when the RECALL@100 decreases 14% while promoting a reduction of 33% in the POS@100. In the case of the bias originated from uneven popularity associated with 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 associated with popularity bias was observed for λ equals to 50. In the first case, RECALL@100 had its value decreased by 24%, 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 effect of removing bias from CF recommendations, we selected a random user, and we show the first 10 predicted scores before and after applying the new term responsible for bias removal. The comparison is shown in

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

Figs. 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 Fig. 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 shown in Fig. 4.

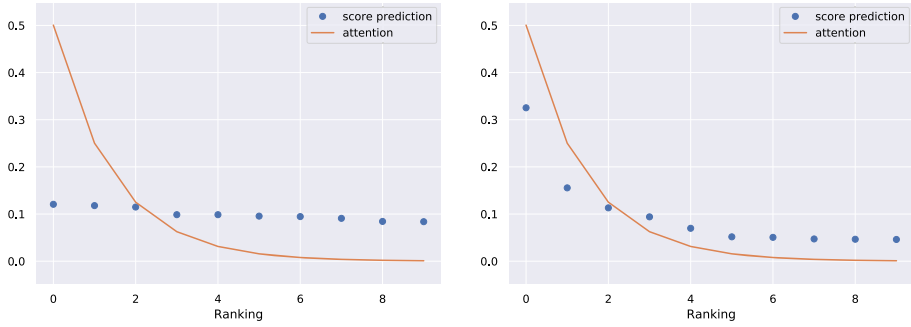


Fig. 3 Top 10 scores calculated for a random user before (Left) and after (Right) applying the new term for removing Position Bias. The attention (solid line) is calculated by a theoretical model, and the predictions are plotted as dots

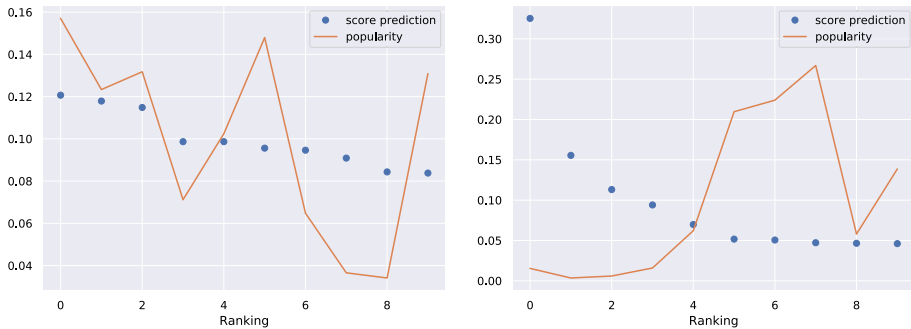


Fig. 4 Top 10 scores calculated for a random user before (Left) and after (Right) applying the new term for removing Popularity Bias. The popularity (solid line) is calculated by summing the ratings an item received in the training subset, and the predictions are plotted as dots

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 or items 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/item interactions. We argue that when that is the case, there is still space for biases and unfairness in the results. We then proposed a new criteria for the so-called *Feature-Blind* fairness, and we discussed possible associations with previous definitions of fairness. 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 needs to be removed, and how much accuracy should be preserved. The framework is based on Variational Autoencoders, and it 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 decreasing trend. The effect was observed when setting the parameter responsible for removing bias from the results (λ) with values greater than one, and its manifestation gets stronger as λ increases. In the case of MovieLens, the effect was also observed for values

of λ greater than 100, but these were not presented in the text. These observations led us to consider the hypothesis that different datasets might have different sensibility to the learning rate reduction, and that higher values of λ might require longer training processes, or at least different intervals for reducing the learning rate.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abdollahpouri H, Burke R, Mobasher B (2017) Controlling popularity bias in learning-to-rank recommendation. In: Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, pp. 42–46. ACM. <https://doi.org/10.1145/3109859.3109912>
2. Abdollahpouri H, Burke R, Mobasher B (2019) 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, pp. 413–418. AAAI Press (2019). <https://aaai.org/ocs/index.php/FLAIRS/FLAIRS19/paper/view/18199>
3. Bellogín A, Castells P, Cantador I (2017) Statistical biases in information retrieval metrics for recommender systems. *Inf Retr J* 20(6):606–634. <https://doi.org/10.1007/s10791-017-9312-z>
4. Bennett J, Lanning S, Netflix N (2007) The netflix prize. In: In KDD Cup and Workshop in conjunction with KDD
5. Biega AJ, Gummadi KP, Weikum G (2018) 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, pp. 405–414. ACM. <https://doi.org/10.1145/3209978.3210063>
6. Borges R, Stefanidis K (2019) Enhancing long term fairness in recommendations with variational autoencoders. In: Chbeir R, Manolopoulos Y, Ilari S, Papadopoulos A (eds.) 11th International Conference on Management of Digital EcoSystems, MEDES 2019, Limassol, Cyprus, pp. 95–102. ACM. <https://doi.org/10.1145/3297662.3365798>
7. Chen J, Kallus N, Mao X, Svacha G, Udell M (2019) 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, pp. 339–348. ACM. <https://doi.org/10.1145/3287560.3287594>
8. Craswell N, Zoeter O, Taylor MJ, Ramsey B (2008) An experimental comparison of click position-bias models. In: Najork M, Broder AZ, Chakrabarti S (eds.) Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, pp. 87–94. ACM. <https://doi.org/10.1145/1341531.1341545>
9. Deldjoo Y, Anelli VW, Zamani H, Bellogín A, Di Noia T (2021) A flexible framework for evaluating user and item fairness in recommender systems. *User Modeling and User-Adapted Interaction (UMUAI)*. <https://doi.org/10.1007/s11257-020-09285-1>
10. Dwork C, Hardt M, Pitassi T, Reingold O, Zemel RS (2012) Fairness through awareness. In: Goldwasser S (ed.) Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, pp. 214–226. ACM (2012). <https://doi.org/10.1145/2090236.2090255>
11. Hardt M, Price E, Srebro N (2016) Equality of opportunity in supervised learning. In: Lee DD, Sugiyama M, von Luxburg U, Guyon I, Garnett R (eds.) Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, Barcelona, Spain, pp. 3315–3323. <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 (2017) beta-vae: Learning basic visual concepts with a constrained variational framework. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, Conference Track Proceedings

13. Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08, pp. 263–272
14. Kallus N, Mao X, Zhou A (2020) Assessing algorithmic fairness with unobserved protected class using data combination. In: Hildebrandt M, Castillo C, Celis E, Ruggieri S, Taylor L, Zanfir-Fortuna G (eds.) FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, p. 110. ACM (2020). <https://doi.org/10.1145/3351095.3373154>
15. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: Bengio Y, LeCun Y (eds.) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 2014, Conference Track Proceedings
16. Kingma DP, Welling M (2019) An introduction to variational autoencoders. *Found Trends® Mach Learn* 12(4):307–392. <https://doi.org/10.1561/22000000056>
17. Liang D, Krishnan RG, Hoffman MD, Jebara T (2018) 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, pp. 689–698. ACM. <https://doi.org/10.1145/3178876.3186150>
18. Locatello F, Abbati G, Rainforth T, Bauer S, Schölkopf B, Bachem O (2019) On the fairness of disentangled representations. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, pp. 14584–14597. <http://papers.nips.cc/paper/9603-on-the-fairness-of-disentangled-representations>
19. Louizos C, Swersky K, Li Y, Welling M, Zemel RS (2016) The variational fair autoencoder. In: Bengio Y, LeCun Y (eds.) 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, Conference Track Proceedings
20. Pitoura E, Stefanidis K, Koutrika G (2021) Fairness in rankings and recommendations: An overview. *The VLDB Journal*
21. Steck H (2011) Item popularity and recommendation accuracy. In: Mobasher B, Burke RD, Jannach D, Adomavicius G (eds.) Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, pp. 125–132. ACM. <https://doi.org/10.1145/2043932.2043957>
22. Stratigi M, Nummenmaa J, Pitoura E, Stefanidis K (2020) Fair sequential group recommendations. In: SAC '20: The 35th ACM/SIGAPP Symposium on Applied Computing, online event, [Brno, Czech Republic], pp. 1443–1452. ACM. <https://doi.org/10.1145/3341105.3375766>
23. Tsintzou V, Pitoura E, Tsaparas P (2019) 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, 2019, vol. 2440. <http://ceur-ws.org/Vol-2440/short4.pdf>
24. Turrin R, Quadrana M, Condorelli A, Pagano R, Cremonesi P (2015) 30music listening and playlists dataset. In: RecSys Posters
25. Yao S, Huang B (2017) Beyond parity: Fairness objectives for collaborative filtering. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, pp. 2921–2930. <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 (2017) Fa*ir: A fair top-k ranking algorithm. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, 2017, pp. 1569–1578. ACM. <https://doi.org/10.1145/3132847.3132938>
27. Zemel RS, Wu Y, Swersky K, Pitassi T, Dwork C (2013) Learning fair representations. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013, JMLR Workshop and Conference Proceedings, vol. 28, pp. 325–333. JMLR.org. <http://proceedings.mlr.press/v28/zemel13.html>
28. Zhu Z, Hu X, Caverlee J (2018) Fairness-aware tensor-based recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, pp. 1153–1162. <https://doi.org/10.1145/3269206.3271795>
29. Zhu Z, Wang J, Caverlee J (2020) 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. <https://doi.org/10.1145/3397271.3401177>



Rodrigo Borges obtained a Bachelor and a Master degrees in Electric Engineering from the Federal University of Minas Gerais (Brazil), and he is currently pursuing a joint PhD in Computer Science from University of São Paulo (Brazil) and from Tampere University (Finland). His doctoral studies are focused on machine learning methods for mitigating unfairness in recommendation systems. His research interests include dynamic recommender systems, algorithmic fairness, machine learning, music and audio signal processing.



Kostas Stefanidis is an Associate Professor at the Faculty of Information Technology and Communication Sciences of the Tampere University in Finland, where he also leads the Group on Recommender Systems. He has more than 10 years of experience in different roles at ICS-FORTH in Greece, NTNU in Norway and CUHK in Hong Kong. He got his PhD in personalized data management from the Univ. of Ioannina in Greece. His research interests are in the broader area of big data. His work focuses on personalization and recommender systems, entity resolution, data exploration and data analytics, with a special focus recently on socio-technical aspects in data management like fairness and transparency, and published in more than 100 papers in top-tier conferences and journals. He has been involved in several international and national research projects, and he is also actively serving the scientific community. Currently, he is the General co-Chair of the SWH Workshop, colocated with ISWC, the Program co-Chair of the DOLAP Workshop, colocated with EDBT, and the Proceedings Chair

of VLDB 2023.