Towards Personalized Fairness based on Causal Notion

Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Yongfeng Zhang
Computer Science, Rutgers University, New Brunswick, NJ 08854, USA
{yunqi.li,hanxiong.chen,shuyuan.xu,yingqiang.ge,yongfeng.zhang}@rutgers.edu

ABSTRACT

Recommender systems are gaining increasing and critical impacts on human and society since a growing number of users use them for information seeking and decision making. Therefore, it is crucial to address the potential unfairness problems in recommendations.

Just like users have personalized preferences on items, users' demands for fairness are also personalized in many scenarios. Therefore, it is important to provide *personalized* fair recommendations for users to satisfy their *personalized* fairness demands. Besides, previous works on fair recommendation mainly focus on association-based fairness. However, it is important to advance from associative fairness notions to causal fairness notions for assessing fairness more properly in recommender systems. Based on the above considerations, this paper focuses on achieving personalized counterfactual fairness for users in recommender systems. To this end, we introduce a framework for achieving counterfactually fair recommendations through adversary learning by generating feature-independent user embeddings for recommendation. The framework allows recommender systems to achieve personalized fairness for users while also covering non-personalized situations. Experiments on two real-world datasets with shallow and deep recommendation algorithms show that our method can generate fairer recommendations for users with a desirable recommendation performance.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Information systems** → **Recommender systems**.

KEYWORDS

Personalized Fairness; Counterfactual Fairness; Recommender System; Adversary Learning

ACM Reference Format:
Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Yongfeng Zhang. 2021. Towards Personalized Fairness based on Causal Notion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3404835.3462966

1 INTRODUCTION

Recently, there has been growing attention on fairness considerations in recommendation models [9, 14, 24, 25, 38–40, 63]. The fairness problem in recommender systems—which are known as multi-stakeholder platforms—should be considered from different perspectives, including user-side, item-side or seller-side [14]. Compared with the many fairness research on item- or seller-side [2–4, 32, 56], fairness issues on the user-side has been less studied in recommender systems. One challenge for user-side fairness research—compared to the item-side—is that different users' fairness demands can be different due to their personalized preferences. For example, as shown on Figure.1, some users may be sensitive to the gender and do not want their recommendations to be influenced by this feature, while others may care more about the age feature and are less concerned about gender. As a result, it is important to explore *personalized fairness* in recommendation scenarios. However, many existing works consider fairness on the same set of sensitive features for all users, and personalized fairness demands are largely neglected. To better assess the unfairness issues in recommendation, it is important to enhance fairness from the personalized view.

Figure 1: Example of users' personalized fairness demands.

Besides, existing work about achieving fairness in recommendations are mainly based on association-based fairness notions, which primarily focus on discovering the discrepancy of statistical metrics between individuals or sub-populations. For example, equalized odds [29], which is one of the most basic criteria for fairness, requires that the false positive rate and true positive rate should be equal for protected group and advantaged group. However, recent research show that fairness cannot be well assessed only based on association notions [33, 37, 67, 68]. A classic example is the Simpson's paradox [49], where the statistical conclusions drawn from the sub-populations and the whole population can be different. In the context of fairness modeling, association-based notions cannot reason about the causal relations between the protected features and the model outcomes. Different from association-based notions, causal-based notions leverage prior knowledge about the world structure in the form of causal models, and thus can help to understand the propagation of variable changes in the system. Therefore, causal-based fairness notions are more and more important to addressing discrimination in machine learning models [33, 37, 67, 68]. In this paper, except for the above mentioned personalized view, we also consider fairness in recommendation from a causal view.

To realize personalized *and* causal fairness for recommendation, we pursue personalized counterfactual fairness in this paper. Counterfactual fairness is an individual-level causal-based fairness notion [37], which considers a hypothetical world beyond the real world.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3462966

To enable fairness, it requires the probability distributions of the model outcomes to be the same in the factual and the counterfactual world for each individual. This individual-level view makes counterfactual fairness a nice fit for both personalized fairness demands and causal fairness notions. In this paper, we expect a recommender system to be counterfactually fair if the recommendation results for a user are unchanged in the counterfactual world where the user's features remain the same except for certain sensitive features specified by the user. This is to grant users with the right to tell us which features—such as gender, race, age, etc.—that they care about and that they want their recommendations to be irrelevant to.

Technically, we introduce a framework for generating recommendations that are independent from the sensitive features so as to meet the counterfactual fairness requirements. We first analyze how we can guarantee the independence between sensitive features and recommendation outcomes. Specifically, we control the dependence between sensitive features and the user embeddings based on the causal graph of the general recommendation pipeline. To generate feature-independent user embeddings, we introduce an adversary learning approach to remove the information of the sensitive features from the user embeddings, while keeping the embeddings informative enough for the recommendation task. To achieve personalized fairness, we allow each user to select a set of sensitive features that they care about. Finally, we provide two methods—the Separate Method (SM) and the Combination Method (CM)—for generating personalized fair recommendations conditioned on the user's sensitive features. Our experiments on two real-world datasets with different types of shallow or deep recommendation algorithms show that our method is able to enhance personalized counterfactual fairness in recommendation with a desirable recommendation performance.

The key contributions of this paper are as follows:

- We consider unfairness issues in recommendation from a personalized perspective to achieve personalized fairness.
- We consider unfairness issues in recommendation from a causal perspective to achieve counterfactual fairness.
- We introduce a framework for achieving personalized counterfactual fairness in recommendation based on adversarial learning.
- We conduct experiments on two real-world datasets with both shallow and deep models to show the effectiveness of our framework on enhancing fairness in recommendation.

In the following, we review related work in Section 2. Before we introduce the method, we provide some preliminaries and notations in Section 3. In Section 4, we introduce the details of our framework. Experimental settings and results are provided in Section 5. Finally, we conclude this work in Section 6.

2 RELATED WORK

2.1 Association-based Fairness

Fairness is becoming more and more important in machine learning [31, 53, 56]. Overall, there are two basic frameworks for algorithmic fairness, i.e., group fairness and individual fairness. Group fairness requires that the protected group and advantaged group should be treated similarly [52], while individual fairness requires that similar individuals are treated similarly. Individual fairness is relatively more difficult to precisely define due to the lack of agreement on similarity metrics for individuals in different tasks [21].

The first endeavor to achieve fairness in the community is to develop association-based (or correlation-based) notions for fairness, which aims to find the discrepancy of statistical metrics between individuals or sub-populations. More specifically, early works about fairness are mostly on classification tasks, which design algorithms that are compatible with fairness constraints [59, 66]. For binary classification, fairness metrics can be expressed by rate constraints, which regularize the classifier's positive or negative rates over different protected groups [19, 46]. For example, demographic parity requires that the classifier's positive rate should be the same across all groups. To achieve fairness, the training objective is usually optimized together with such constraints over fairness metrics [6, 26].

Some recent works have also considered the fairness of ranking tasks. Some works directly learn a ranking model from scratch [40, 47, 57, 65], while others consider re-ranking or post-processing algorithms for fair ranking [11, 15, 39]. The fairness metrics for ranking tasks are usually defined over the exposure of items that belong to different protected groups. As summarized in [47], such metrics include the unsupervised criteria and the supervised criteria. Unsupervised criteria posit that the average exposure at the top of the ranking list is equal for different groups [15, 56, 65], while the supervised criteria require the average exposure of item groups to be proportional to their average relevance to the query [11, 57].

Recommendation algorithm can usually be considered as a type of ranking algorithms. However, it is also special in that personalization is a very fundamental consideration for recommendation. As a result, different from previous fairness ranking algorithms which usually consider fairness from the item-side, we also need to consider fairness on the user-side in recommendation, as well as users' personalized fairness demands.

2.2 Causal-based Fairness

Recently, researches have noticed that fairness cannot be well assessed merely based on correlation or association [33, 37, 67, 68], since they cannot reason about the causal relations between input and output. However, real discrimination may result from a causal relation between the model decisions (e.g. hiring and admission) and the sensitive features (e.g. gender and race). Therefore, causal-based fairness notions are proposed. Causal-based fairness notions are mostly defined on intervention or counterfactual. Intervention can be achieved through random experiments, while counterfactual considers a hypothetical world beyond the real world. We introduce some important causal notions as follows.

Total effect (TE) [49] measures the effect of changing the sensitive feature along all the causal paths to the outcome. Treatment on the treated (ETT) [49], which is the most basic fairness notion under counterfactuals, measures the difference between the real world and the counterfactual world where the sensitive feature changes for the individual. Both TE and ETT seek the equality of outcomes between protected and unprotected groups. Disparate treatment [8] is another framework which aims at ensuring the equality of treatment by prohibiting the use of sensitive features when making decisions, including direct effect, indirect effect and path-specific effect. Direct discrimination is measured by the causal effect along

the causal path from the sensitive feature to the final decision [50]; indirect discrimination is assessed by the causal effect along the causal path through proxy features [50]; while path-specific effect [49] characterizes the causal effect over specific paths. Based on this, various causal-based notions have been put forward. Examples include: No unresolved discrimination [34], which measures the indirect causal effects from sensitive features to outcomes and requires that there is no directed path from sensitive features to outcomes except via a resolving variable; Equality of effort [30], which measures how much efforts are needed from the protected group or individual to reach a certain level of outcome to identify discrimination; PC-Fairness [60], which can cover lots of causal-based fairness notions by tuning its parameters. A more comprehensive list of causal-based fairness notions are provided in [42].

This paper aims at achieving counterfactual fairness in recommendation. Counterfactual fairness [37] is a fine-grained variant of ETT conditioned on all features. It requires the probability distribution of the outcome to be the same in the factual and counterfactual worlds for every individual. We will introduce the definition of counterfactually fair recommendation in detail in the preliminaries.

2.3 Fair Recommendation

Different from fair classification and ranking, the concept of fairness in recommendation can be more complex as it extends to multiple stakeholders [14]. Recent works on fairness in recommendations have very different views. Lin et al. [40] introduced an optimization framework for fairness-aware group recommendation based on Pareto Efficiency. Yao and Huang [63] explored fairness in collaborative filtering recommender systems, which proposed four metrics to assess different types of fairness by adding fairness constraints to the learning objective. Burke [14] and Abdollahpouri and Burke[1] categorized different types of multi-stakeholder platforms and introduced several desired group fairness properties. Leonhardt et al. [38] identified the unfairness issue for users in post-processing algorithms to improve the diversity in recommendation. Mehrotra et al. [43] proposed a heuristic strategy to jointly optimize fairness and performance in two-sided marketplace platforms. Beutel et al. [9] considered fairness in recommendation under a pairwise comparative ranking framework, and offered a regularizer to improve fairness when training recommendation models. Patro et al. [48] explored individual fairness for both producers and customers for long-term sustainability of two-sided platforms. Fu et al. [24] impaired the group unfairness problem in the context of explainable recommendation [70, 71] over knowledge graphs; Li et al. [39] considered user-oriented fairness in recommendation by requiring the active and inactive user groups be treated similarly; Ge et al. [25] proposed a reinforcement learning framework to deal with the changing group labels of items to achieve long-term fairness in recommendation. To the best of our knowledge, our work is the first to consider personalized and causal-based fairness in recommender systems.

3 PRELIMINARIES AND NOTATIONS

In this section, we introduce the preliminaries and notations used in this paper. Capital letters such as Z denote variables, lowercase letters such as z denote specific values of the variables. Bold capital letters such as $\mathbf{Z}$ denote a set of variables, while bold lowercase letters such as $\mathbf{z}$ denote a set of values. In the following, we first show the notations used in the recommendation task, and then we introduce the preliminaries about counterfactual fairness.

3.1 Recommendation Task

In recommendation task [16], we have a user set $\mathbb{U} = \{u_1, u_2, \cdots, u_n\}$ and an item set $\mathbb{V} = \{v_1, v_2, \cdots, v_m\}$, where n is the number of users and m is the number of items. The user-item interaction histories are usually represented as a 0-1 matrix $H = \left[h_{ij}\right]_{n\times m}$, where each entry $h_{ij} = 1$ if user u_i has interacted with item v_j, otherwise $h_{ij} = 0$. The key task for recommendations is to predict the preference scores of users over items, so that the model can recommend each user u_i a top-N recommendation list $\{v_1, v_2, \cdots, v_N|u_i\}$ according to the predicted scores. To learn the preference scores, modern recommender models are usually trained to learn the user and item representations based on the user-item interactions, and then take the representations as input to a learned or designed scoring functions to make recommendations. We use $\mathbf{r}_u$ and $\mathbf{r}_v$ to represent the learned vector embeddings for user u and item v, and use S_{uv} to denote the predicted preference score for a (u, v) pair. In addition to the interaction records, users have their own features, such as gender, race, age, etc. In particular, we use $\mathbf{Z}$ to represent the sensitive features, and use $\mathbf{X}$ to denote all the remaining features which are not causally dependent on $\mathbf{Z}$, i.e., the insensitive features. Without loss of generality, we suppose each user have K categorical sensitive features $\{Z_1, Z_2, ..., Z_K\}$.

3.2 Counterfactual

Before introducing the definition of counterfactual fairness, we first briefly introduce the concept of counterfactual. To understand counterfactual, let us consider an example first [51]. When Alice was driving home, she came to a fork in the road and had to make a choice: to take the street 1 ($X = 1$) or to take the street 0 ($X = 0$). Alice took the street 0 and it took her 2 hours to arrive home, and then she may ask "how long would it take if I had taken the street 1 instead?" Such a "what if" statement in which the "if" portion is unreal or unrealized, is known as a counterfactual [51]. The "if" portion of a counterfactual is called the antecedent. We use counterfactual to compare two outcomes under the exact same condition, differing only in the antecedent. To solve the above counterfactual, we denote the driving time of the street 1 by $Y_{X=1}$ or Y_1, and the driving time of the street 0 by $Y_{X=0}$ or Y_0, then the quantity we want to estimate is $E(Y_{X=1}|X = 0, Y = Y_0 = 2)$. The counterfactual can be solved based on structural causal model [49]. As the assumption of causal models, the state of Y will be fully determined by the background variables U and the structural equations F. Specifically, given $U = u$ (which can be derived from the evidence of $X = 0$ and $Y = 2$), and an intervention on X as $do(X = 1)$, we can derive the solution of the counterfactual. To make the notations clear, we use the expression $P(Y_{X=x'} = y'|X = x, Y = y) = P(y'_{x'}|x, y)$, which involves two worlds: the observed world where $X = x$ and $Y = y$ and the counterfactual world where $X = x'$ and $Y = y'$. The expression reads "the probability of $Y = y'$ had X been x' given that we observed $Y = y$ and $X = x$".

3.3 Counterfactual Fairness

Counterfactual fairness is an individual-level causal-based fairness notion [37]. It requires that for any possible individual, the predicted result of the learning system should be the same in the counterfactual world as in the real world. The counterfactual world here is the world in which we only make an intervention on user's sensitive features, while all other features of the user that are not dependent on the sensitive features are kept unchanged. The counterfactual fairness is an individual-level notion because it is conditioned on all the unchanged variables. Here is an example for counterfactual fairness: suppose we are designing a decision making system that does not discriminate against gender when deciding students' admission to college. Counterfactual fairness requires that the admission result to a student will not be changed if his or her gender were reversed while all other features that are not dependent on gender remain the same, such as grades and recommendation letters. Such a fairness requirement is usually more reasonable than forcefully requiring the same admission rate for all genders in association-based notions, since students of different genders may have different preferences for college majors.

In this paper, we consider counterfactual fairness in recommendation scenario. We give the definition of counterfactually fair recommendation as follows.

Definition 1 (Counterfactually fair recommendation). *A recommender model is counterfactually fair if for any possible user u with features $\boldsymbol{X} = \boldsymbol{x}$ and $\boldsymbol{Z} = \boldsymbol{z}$:*

$$P\left(L_{\boldsymbol{z}} \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Z} = \boldsymbol{z}\right) = P\left(L_{\boldsymbol{z}'} \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Z} = \boldsymbol{z}\right)$$

for all L and for any value $\boldsymbol{z}'$ attainable by $\boldsymbol{Z}$, where L denotes the Top-N recommendation list for user u.

Here $\mathbf{Z}$ are the user's sensitive features and $\mathbf{X}$ are the features that are not causally dependent on $\mathbf{Z}$. This definition requires that for any possible user, sensitive features $\mathbf{Z}$ should not be a cause of the recommendation results. Specifically, for a given user u, the distribution of the generated recommendation results L for u should be the same if we only change $\mathbf{Z}$ from $\mathbf{z}$ to $\mathbf{z}'$, while holding the remaining features $\mathbf{X}$ unchanged.

4 FAIR RECOMMENDATION FRAMEWORK

In this section, we introduce the framework for generating counterfactually fair recommendations.

4.1 Problem Formulation

As discussed above, we aim at achieving counterfactual fairness in recommendations. From Definition 1, we can see that counterfactual fairness requires that the generated recommendation results L are independent from the user sensitive features $\mathbf{Z}$. As stated in [37], which considers counterfactual fairness in classification tasks, the most straightforward way to guarantee the independence between predicted outcomes and sensitive features is just avoiding from using sensitive features (and the features causally depend on the sensitive features) as input. However, this is not the case in recommendation scenarios. Most of the Collaborative Filtering (CF) [22, 27] or Collaborative Reasoning (CR) [16, 55] recommender systems are directly trained from user-item interaction history, and content-based recommendation models [5, 41, 69] and hybrid models [13] may use user profiles as input or use additional information to help train the model. However, no matter if the model directly uses user features as input or not, the model may generate unfair recommendations on some user features. The reason is that by collaborative learning (CF or CR) in the training data, the model may capture the relevance between user features and user behaviours that are inherently encoded into the training data, since user features may have causal impacts on user behaviors and preferences. As a result, we need to design methods to achieve counterfactually fair recommendations as it cannot be realized in trivial way.

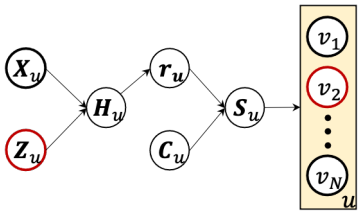


Figure 2: Causal relations for general recommendation models. For a given user u, $\mathbf{X}_u$ and $\mathbf{Z}_u$ are insensitive and sensitive features of u, respectively. $\mathbf{H}_u$ is the user interaction history. $\mathbf{r}_u$ is the user embedding. $\mathbf{C}_u$ is the candidate item set for u. $\mathbf{S}_u$ are the predicted scores over the candidate items. The red circled nodes are used to emphasize the impact of the sensitive features on the final recommendation list.

To guarantee that recommendation results are independent from user sensitive features, we only need to require that given a user u, for any item $v \in \mathcal{V}$, the predicted score S_{uv} for the user-item pair (u, v) is independent from the user sensitive features $\mathbf{Z}$. As shown in Figure 2, which represents the causal relations for general recommendation models, for a given user u, the scoring function $\mathbf{S}_u$ usually takes user embedding $\mathbf{r}_u$ and candidate item embeddings $\mathbf{C}_u$ as input to generate the recommendation list. However, the user embedding $\mathbf{r}_u$, which is learned from user histories $\mathbf{H}_u$, may depend on the user features $\mathbf{X}_u$ and $\mathbf{Z}_u$ since the features causally impact user behaviours. Therefore, as shown by the causal path from sensitive feature $\mathbf{Z}_u$ to the final recommendation result, we only need to ensure the independence between user embedding $\mathbf{r}_u$ and the sensitive feature $\mathbf{Z}_u$ to meet the counterfactual fairness requirement, i.e., for all $u \in \mathcal{U}$, we need to guarantee $\mathbf{r}_u \perp \mathbf{Z}_u$.

Besides, to meet users' personalized demands on fairness, we allow each user to select a set of sensitive features that they care about, and we generate fair recommendations in terms of these features. Suppose user u selected a set of sensitive features $\mathbf{Q}_u \subseteq \{1, \ldots, K\}$, then we need to guarantee that $\mathbf{r}_u \perp z_u^k$, for all $k \in \mathbf{Q}_u$.

4.2 The Model

In this section, we introduce the model to generate feature independent user embeddings through adversary learning. The main idea is to train a predictor and an adversarial classifier simultaneously, where the predictor aims at learning informative representations for the recommendation task, while the adversarial classifier aims at minimizing the predictor's ability to predict the protected features from the learned representation, and thus the information about sensitive features are removed from the representations to mitigate

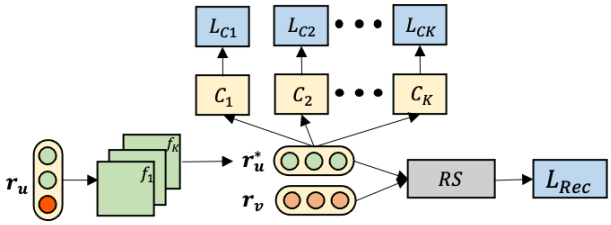


Figure 3: The architecture of our framework. For a given user u with the original representation $\mathbf{r}_u$, we first use the filter module to remove the information about sensitive features and get the filtered embedding $\mathbf{r}_u^*$. Then we use the corresponding classifier C_k to predict the k-th sensitive feature from the filtered embedding, and on the other hand, we train a recommender system (RS) for the main task. The loss of recommendation and classification are optimized together.

discrimination [7, 10, 12, 20, 23, 58, 61]. Following the adversary learning setup [7, 12, 28], we develop an adversary network that consists two modules: a filter module that aims at filtering out the information about sensitive features from user embeddings, and a discriminator module that aims to predict the sensitive features from the learned user embeddings. Figure 3 shows the architecture of our method.

4.2.1 ***Filter Module.*** Given a learning algorithm that learns user embedding $\mathbf{r}_u$ to generate recommendations for user u, we require the embedding $\mathbf{r}_u$ to be independent from certain user features to achieve counterfactual fairness. Therefore, we first introduce a filter module with a set of filter functions, which are used to filter out the information about certain sensitive features in the user embeddings. We denote the filter function as $f : \mathbb{R}^d \mapsto \mathbb{R}^d$, and the filtered embedding $f(\mathbf{r}_u)$ is independent from certain sensitive features while maintaining other insensitive information of the user. To meet users' personalized fairness demands, we allow each user to select a set of sensitive features $\mathbf{Q}_u \subseteq \{1, \ldots, K\}$. To achieve personalized fairness in recommendation, we provide two methods as follows.

In most recommendation scenarios, such as in movie, music, and e-commerce recommendation, users are not willing to share too much personalized information about themselves with the system, and they may only select a few sensitive features for generating fair recommendations, i.e., K will be a very small number. In this scenario, a straightforward way is to train one filter function for each potential combination of the sensitive features. For example, if $K = 2$, and $\mathbf{Q}_u$ contains two sensitive features *Age* and *Gender*, then we need to train filter functions $f_A, f_G, f_{A,G}$ to remove the sensitive information of *Age*, *Gender*, and both *Age* and *Gender*, from the user embeddings, respectively. We call this method the Separate Method (SM). The architecture of separate method of this example is show in Figure.4. We denote the filtered embedding of user u in terms of the selected sensitive feature set $\mathbf{Q}_u$ as follows.

$$\mathbf{r}_u^* = f_{\mathbf{Q}_u}(\mathbf{r}_u)$$

However, in some cases such as social network recommendation, users may have many sensitive features to consider, and the potential combinations of sensitive features will be quite a lot. Under such scenarios, training one filter for each combination is infeasible due

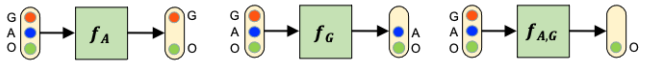


Figure 4: The architecture of Separate Method. G represents *Gender*; A represents *Age*; O represents *Others.*

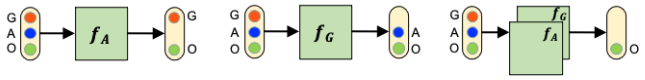


Figure 5: The architecture of Combination Method. G represents *Gender*; A represents *Age*; O represents *Others.*

to the exponential number of combinations. Therefore, we introduce the Combination Method (CM) to achieve personalized fairness in recommendation. Specifically, we train one filter function for each sensitive feature. The filter functions $\{f_1, f_2, ..., f_K\}$ correspond to sensitive features $\{Z_1, Z_2, ..., Z_K\}$, where f_k is trained to filter the information about Z_k. Considering the above example where $\mathbf{Q}_u$ contains two sensitive features *Age* and *Gender*, the architecture of combination method is shown in Figure.5. To generate the feature independent embedding of user u with $\mathbf{Q}_u = \{q_1, q_2, \ldots, q_{|\mathbf{Q}_u|}\}$, we can simply combine the $|\mathbf{Q}_u|$ filtered embeddings as:

$$\mathbf{r}_u^* = G(f_{q_1}(\mathbf{r}_u), f_{q_2}(\mathbf{r}_u), \ldots, f_{q_{|\mathbf{Q}_u|}}(\mathbf{r}_u))$$

where G is a combination function that takes the $|\mathbf{Q}_u|$ filtered embeddings as input, and outputs the embedding which is independent from all the features in $\mathbf{Q}_u$ without changing the embedding dimension. For example, we can simply use the average of the $|\mathbf{Q}_u|$ filtered embeddings as G. We will compare the performance of the SM and CM methods in the experiment section.

Furthermore, for non-personalized situation where the fairness demands of users are the same—e.g., all the users ask for fair recommendations on race—we can simply train one filter function corresponding to the sensitive features.

4.2.2 ***Discriminator Module.*** To learn filter functions, we use the idea of adversary learning to train a set of discriminators. Specifically, for each sensitive feature Z_k, we train a classifier $C_k : \mathbb{R}^d \mapsto [0,1]$, which attempts to predict Z_k from the user embeddings. The goal of the filter functions is to make it hard for the classifiers to predict the sensitive features from the user embeddings, while the goal of the discriminators is to fail the filter functions. Concretely, the training process tries to jointly optimize both goals.

4.2.3 ***Adversary Training.*** We use $\mathcal{L}_{Rec}$ to denote the loss of the recommendation task. Depending on the recommendation model, $\mathcal{L}_{Rec}$ can be the pair-wise ranking loss [54] or mean square error loss [36], etc. We use $\mathcal{L}_C$ to denote the loss of the discriminators, i.e., the loss of the classification task, which is a cross-entropy loss in our implementation. We thus define our adversary learning loss as follows:

$$\mathcal{L} = \sum_{u,v,\mathbf{Q}_u} \left(\mathcal{L}_{\text{Rec}}(u, v, \mathbf{Q}_u) - \lambda \sum_{k \in \mathbf{Q}_u} \sum_{z_k \in Z_k} \mathcal{L}_C\left(\mathbf{r}_u^*, z_k\right) \right) \quad (1)$$

where adversarial coefficient λ controls the trade-off between recommendation performance and fairness. We study the influence of λ in the ablation study section. The adversary learning algorithm is shown in Algorithm 1.

We also provide the following theorem to show the theoretical guarantee that the adversary training procedure can make the filtered user embeddings independent from the sensitive features.

Theorem 4.1. *If (1) the filter functions and discriminators are implemented with sufficient capacity, and (2) at each step of Algorithm 1, the discriminators are allowed to reach their optimum given the filter functions, and (3) the filter functions are optimized according to the loss function with discriminators fixed, then we have for any user u, $\boldsymbol{r}_u \perp \boldsymbol{Z}_u$ as $\lambda \to \infty$.*

Proof. For simplicity, we consider the case of a single binary sensitive feature, i.e., we have one sensitive feature Z which can be 0 or 1. In this case, the loss of the discriminators in Eq.(1)—which will be a binary cross-entropy loss—is the same as the loss of Generative Adverserisal Networks (GAN) [28]. According to the Proposition 2 of [28], it has been proven that if (1) the generator and discriminator have enough capacity, (2) the discriminator is allowed to reach the optimum during the training process, and (3) the generator is updated with the discriminator fixed so as to improve the criterion, then the distribution of the fake data will converge to the distribution of the real data. As a result, when classifying a binary sensitive feature, we have that the distributions of user embeddings with sensitive feature $Z = 0$ and $Z = 1$ will be the same once converged, i.e., we have $P(\mathbf{r}_u^*|Z = 0) = P(\mathbf{r}_u^*|Z = 1)$, which gives $\mathbf{r}_u^* \perp Z$. □

The theoretical intuition above can be generalized to the multi-class and multi-feature settings. Cho et al. [18] have theoretically shown how to optimize the independence between predictions and sensitive features in multi-settings from a mutual information perspective, which leads to the same result in Theorem 4.1.

4.3 Training Algorithm

For adversary learning, we adopt mini-batch training in our implementation. Specifically, for each batch, we first feed the input to the model to obtain $\mathcal{L}_{Rec}$ and $\mathcal{L}_C$, and then we fix the parameters in the discriminator and optimize the recommendation model as well as the corresponding filter functions by minimizing $\mathcal{L}$. After that, the parameters of the recommendation model and filter functions are fixed, and $\mathcal{L}_C$ is minimized for t steps. Here $t = 10$ in our implementation.

To achieve personalized fairness, we allow each user to select a set of concerned sensitive features $\mathbf{Q}_u \subseteq \{1, \ldots, K\}$. In implementation of the learning algorithm, we sample a binary mask for each batch to determine $\mathbf{Q}_u$ to train the filtered embedding $\mathbf{r}_u^*$. Specifically, the binary mask is sampled from K independent Bernoulli distributions with the probability $p = 0.5$ under the assumption that there is no causal relation between sensitive features when user selects them. When there is a need to consider the dependence between sensitive features, we can simply apply other distributions based on different applications. Again, the pseudo-code for the entire training algorithm is given in Algorithm 1.

Algorithm 1: Adversarial Training Algorithm

Input : Training user set $\mathcal{U}$; item set $\mathcal{V}$; Recommendation model RS; Filter functions $\mathcal{F}$; Discriminators C; Sensitive features $\mathbf{Z}$; Training epochs M; Discriminator training steps T; Adversarial coefficient λ;

Initialize: user embeddings $\mathbf{r}_u, \forall u \in \mathcal{U}$, item embeddings $\mathbf{r}_v, \forall v \in \mathcal{V}$;
for *epoch* $\leftarrow 1$ **to** M **do**
  for $u \in \mathcal{U}, v \in \mathcal{V}$ **do**
    $\mathbf{Q} \leftarrow$ sample binary filter mask;
    if *Separate Method* **then**
      $F \leftarrow$ get filter function $f_{\mathbf{Q}}$ from $\mathcal{F}$;
    end
    if *Combination Method* **then**
      $F \leftarrow$ get filter functions $\{f_k\}_{k \in \mathbf{Q}}$ from $\mathcal{F}$;
    end
    $\{C_k\}_{k \in \mathbf{Q}} \leftarrow$ get discriminators from C;
    $\{z_k\}_{k \in \mathbf{Q}} \leftarrow$ get feature values from $\mathbf{Z}$;
    $\mathbf{r}_u^* \leftarrow F(\mathbf{r}_u)$ ▷ obtain filtered user embedding;
    $\mathcal{L}_{Rec} \leftarrow \mathrm{RS}(\mathbf{r}_u^*, \mathbf{r}_v)$;
    $\mathcal{L}_C \leftarrow \sum_{k \in \mathbf{Q}} C_k(\mathbf{r}_u^*, z_k)$
    $\mathcal{L} \leftarrow \mathcal{L}_{Rec} + \lambda \mathcal{L}_C$;
    Optimize $\mathcal{L}$ *w.r.t* $\mathbf{r}_u, \mathbf{r}_v, F$, RS, with $\{C_k\}_{k \in \mathbf{Q}}$ fixed;
    for $t \leftarrow 1$ **to** T **do**
      $\mathcal{L}_C \leftarrow \sum_{k \in \mathbf{Q}} C_k(\mathbf{r}_u^*, z_k)$;
      Optimize $\mathcal{L}_C$ *w.r.t* $\{C_k\}_{k \in \mathbf{Q}}$ with $\mathbf{r}_u, F$ fixed;
    end
  end
end

5 EXPERIMENTS

In this section, we first briefly introduce the datasets, baselines and experimental setup used for the experiments. Then we show and analyze the main experimental results, including the comparison of recommendation performance and fairness between the baseline models and the two fairness models. Finally, we conduct ablation studies to further analyze the algorithm.

5.1 Dataset Description

To evaluate the models under different data scales, data sparsity and application scenarios, we perform experiments on a movie recommendation dataset and an insurance recommendation dataset, which are two real-world and publicly available datasets.

MovieLens[1]. We use the MovieLens-1M dataset which contains user-item interactions and user profile information for movie recommendation. We select *gender*, *age* and *occupation* as user sensitive features, where *gender* is a binary feature, *occupation* is a 21-class feature, and for *age*, we assign users into 13 equal length groups based on their age range.

Insurance[2]. This is a Kaggle dataset with the goal of recommending insurance products to a target user. For each user, we select *gender*, *marital_status* and *occupation* as sensitive features, where *gender* is still a binary feature. For *marital_status* and *occupation*,

[1] https://grouplens.org/datasets/movielens/1m/
[2] https://www.kaggle.com/mrmorj/insurance-recommendation

we group up the minority classes to transform them into 3-class features due to the severe data imbalance over classes. To guarantee the data quality for training models, we filter out the users with less than 4 interactions to make a denser dataset.

The statistics of the datasets are summarized in Table 1. In our experiments, we split each dataset into train (80%), validation (10%) and test sets (10%) and all the baseline models share these datasets for training and evaluation.

5.2 Evaluation Methods

We consider standard metrics Normalized Discounted Cumulative Gain (NDCG@N) and Hit rate (Hit@N) scores to evaluate the top-N recommendation quality. For the MovieLens dataset, we report NDCG@5 and Hit@5. For the Insurance dataset, we show NDCG@3 and Hit@3 scores due to the limited number of candidates in this dataset. For efficiency consideration, we use sampled negative interactions for evaluation instead of computing the user-item pair scores for each user over the entire item space [72]. For each user, we randomly select 100 negative samples that the user has never interacted with. These negative items are put together with the positive item in the validation or test set to constitute the user's candidates list. Then we compute the metric scores over this candidates list to evaluate the recommendation model's top-N ranking performance. The result of all metrics in our experiments are averaged over all users.

Following the settings in learning fair representations by adversary learning such as [7, 12, 23], to evaluate the effectiveness of discriminators, we train a set of attackers which have totally the same structure and capacity as the discriminators. Specifically, after we finish training the main algorithm, we input the filtered user embeddings and their corresponding sensitive labels to the attackers so as to train them to classify the sensitive features from the filtered embeddings. Just as the discriminators, we train one attacker for each sensitive feature. If the attackers can distinguish sensitive features from the user embeddings, then we say that sensitive features are leaked into the user embeddings, thus the recommendation model is not counterfactually fair.

For training and evaluating attackers, we split the data into train (80%) and test sets (20%). We report AUC score for each attacker to show if the filtered user embeddings can be classified correctly by the attacker. For multi-class evaluation, we calculate the AUC score for all the combination of feature pairs and apply their macro-average to make the result insensitive to imbalanced data. The AUC score falls into the range of [0.5,1], the lower the better. An ideal result to meet the counterfactual fairness requirement is an AUC score of about 0.5, which means the attacker cannot guess the sensitive feature out of the user embeddings at all.

5.3 Baselines

In order to evaluate the effectiveness of our proposed framework, we apply our method over both shallow and deep recommendation models. We introduce the baseline models as follows:

- **PMF** [45]: The Probabilistic Matrix Factorization algorithm by adding Gaussian prior into the user and item latent factor distributions for matrix factorization.

Table 1: Statistics of the datasets

Dataset	#Interactions	#Users	#Items	Sparsity
MovieLens	1,000,209	6,040	3,952	95.81%
Insurance	5,382	1,231	21	79.18%

Table 2: AUC scores of all attackers on the MolveLens and Insurance datasets. G, A, O represent *gender*, *age* and *occupation* respectively on MovieLens; while G, M, O represent *gender*, *marital_status* and *occupation* respectively on Insurance. The best results are highlighted in bold.

		MoiveLens			Insurance		
		AUC-G	AUC-A	AUC-O	AUC-G	AUC-M	AUC-O
PMF	Orig.	0.7697	0.8428	0.6024	0.6253	0.7098	0.6577
	SM	**0.5389**	**0.5560**	**0.5289**	**0.5340**	**0.5377**	**0.5492**
	CM	0.5532	0.5951	0.5396	0.5419	0.5789	0.5540
BiasedMF	Orig.	0.7870	0.8403	0.6064	0.6183	0.7715	0.6357
	SM	**0.5345**	**0.5601**	**0.5258**	**0.5000**	**0.5405**	**0.5555**
	CM	0.5519	0.5757	0.5300	0.5491	0.5430	0.5717
DeepModel	Orig.	0.7165	0.7571	0.5481	0.5952	0.6339	0.6086
	SM	0.5545	**0.5833**	0.5445	**0.5202**	**0.5687**	0.5815
	CM	**0.5371**	0.6075	**0.5247**	0.5335	0.5765	**0.5407**
DMF	Orig.	0.7049	0.7238	0.5710	0.6172	0.6309	0.6023
	SM	0.6073	0.5670	0.5289	0.5421	**0.5638**	**0.5653**
	CM	**0.5000**	**0.5297**	**0.5120**	**0.5258**	0.5873	0.5791

- **BiasedMF** [36]: A matrix factorization algorithm which takes user and item latent factors as well as the global bias terms into consideration.
- **DeepModel** [17]: This algorithm applies deep neural network with non-linear activation functions to train a user and item matching function.
- **DMF** [62]: Deep Matrix Factorization is a deep model for recommendation, which uses multi-layer perceptron with non-linear activation function to encode the raw user-item interaction matrix into dense latent factor representations.

5.4 Experimental Settings

To better accommodate the ranking task, we apply the Bayesian Personalized Ranking (BPR) [54] loss as the recommendation loss in Eq.(1) for all the baseline models. For each user-item pair in the training dataset, we randomly sample one item that the user has never interacted with as the negative sample in one training epoch. We set the learning rate to 0.001. ℓ_2-regularization coefficient is 0.0001 for all the datasets. Dropout rate is 0.2. Early stopping is applied and the best models are selected based on the performance on the validation set. Rectified Linear Unit (ReLU) is used as the activation function for DMF and DeepModel. We apply Adam [35] as the optimization algorithm to update the model parameters. The adversarial coefficient λ in Eq.(1) is selected from $[10, 20, 50]$ for MovieLens, while $[100, 200, 500, 1000]$ for the Insurance dataset. The filter modules are two-layer neural networks with LeakyReLU as the non-linear activation function. The classifiers (discriminators and attackers) are multi-layer perceptrons with the number of layers

Table 3: The recommendation performance of baselines, Separate Method (SM) and Combination Method (CM) on MovieLens. Orig. represents the baseline model; G, A, O represent *gender*, *age* and *occupation*, respectively. The performance of SM and CM are evaluated for all combinations of the three sensitive features. For example, SM-G represents the performance of filtering user embeddings by f_{gender} trained via SM. The better results between SM and CM are highlighted in bold.

		Orig.	SM-G	CM-G	SM-A	CM-A	SM-O	CM-O	SM-GA	CM-GA	SM-GO	CM-GO	SM-AO	CM-AO	SM-GAO	CM-GAO
PMF	N@5	0.4961	**0.4801**	0.4604	**0.4781**	0.4596	**0.4751**	0.4605	**0.4730**	0.4673	**0.4737**	0.4685	0.4674	**0.4681**	0.4599	**0.4705**
	H@5	0.6493	**0.6342**	0.6179	**0.6318**	0.6188	**0.6291**	0.6191	**0.6273**	0.6251	**0.6282**	0.6274	0.6207	**0.6265**	0.6165	**0.6289**
BiasedMF	N@5	0.4960	**0.4776**	0.4649	**0.4740**	0.4665	**0.4748**	0.4672	0.4710	**0.4732**	0.4699	**0.4742**	0.4672	**0.4744**	0.4573	**0.4767**
	H@5	0.6471	**0.6305**	0.6205	**0.6270**	0.6220	**0.6285**	0.6233	0.6248	**0.6291**	0.6240	**0.6305**	0.6208	**0.6303**	0.6118	**0.6324**
DeepModel	N@5	0.3935	**0.3834**	0.3803	**0.3827**	0.3793	**0.3825**	0.3790	**0.3819**	0.3809	**0.3820**	0.3808	0.3797	**0.3800**	0.3782	**0.3808**
	H@5	0.5501	**0.5370**	0.5338	**0.5357**	0.5325	**0.5357**	0.5325	**0.5349**	0.5343	**0.5350**	0.5343	0.5322	**0.5337**	0.5311	**0.5344**
DMF	N@5	0.3307	**0.3262**	0.3167	**0.3256**	0.3168	**0.3260**	0.3166	**0.3254**	0.3177	**0.3267**	0.3169	**0.3253**	0.3164	**0.3263**	0.3183
	H@5	0.4795	**0.4731**	0.4598	**0.4709**	0.4588	**0.4731**	0.4603	**0.4707**	0.4606	**0.4714**	0.4599	**0.4714**	0.4603	**0.4732**	0.4622

Table 4: The recommendation performance of baselines, Separate Method (SM) and Combination Method (CM) on Insurance. Orig. represents the baseline model; G, M, O represent *gender*, *marital_status* and *occupation*, respectively. The performance of SM and CM are evaluated for all combinations of the three sensitive features. For example, SM-G represents the performance of filtering user embeddings by f_{gender} trained via SM. The better results between SM and CM are highlighted in bold.

		Orig.	SM-G	CM-G	SM-M	CM-M	SM-O	CM-O	SM-GM	CM-GM	SM-GO	CM-GO	SM-MO	CM-MO	SM-GMO	CM-GMO
PMF	N@3	0.6518	**0.6528**	0.6208	**0.6521**	0.6081	**0.6406**	0.6208	**0.6242**	0.6173	**0.6535**	0.6208	**0.6228**	0.6163	**0.6390**	0.6191
	H@3	0.7528	**0.7398**	0.7026	**0.7398**	0.6914	**0.7416**	0.7026	**0.7323**	0.6989	**0.7398**	0.7026	**0.7435**	0.6989	**0.7305**	0.7007
BiasedMF	N@3	0.6209	0.5936	**0.6095**	**0.6190**	0.5995	0.6031	**0.6112**	0.5936	**0.6079**	0.5991	**0.6116**	0.6041	**0.6056**	0.6028	**0.6095**
	H@3	0.7082	0.6803	**0.6952**	**0.7100**	0.6822	0.6877	**0.6952**	0.6803	**0.6952**	0.6803	**0.6970**	**0.6952**	0.6914	0.6859	**0.6952**
DeepModel	N@3	0.6438	**0.6389**	0.6315	**0.6359**	0.6290	**0.6410**	0.6317	**0.6355**	0.6250	**0.6333**	0.6190	**0.6401**	0.6317	**0.6357**	0.6275
	H@3	0.7398	**0.7212**	0.7175	**0.7193**	0.7082	**0.7286**	0.7193	**0.7249**	0.7063	**0.7138**	0.7082	**0.7268**	0.7193	**0.7212**	0.7156
DMF	N@3	0.5301	**0.4988**	0.4751	**0.5122**	0.4927	**0.5108**	0.4858	**0.5143**	0.4731	**0.5115**	0.4827	0.4986	**0.5040**	**0.5231**	0.4652
	H@3	0.6822	0.6283	**0.6301**	**0.6468**	0.6431	**0.6413**	0.6320	**0.6375**	0.6115	**0.6394**	0.6245	**0.6264**	0.6245	**0.6580**	0.6190

set to 7, LeakyReLU as the activation function, and the dropout rate is set to 0.3. Batch normalization is applied for training classifiers.

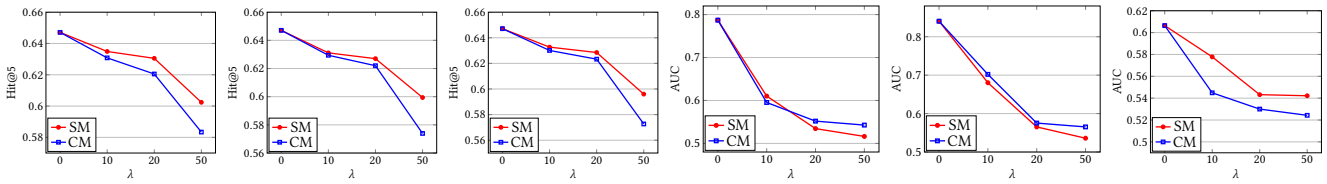
5.5 Main Results

In this section, we show the main results of our experiments, including comparing recommendation performance and fairness for all the baseline models under the SM and CM settings.

Fairness Improvement. We provide the evaluation results of attackers in Table 2 to present the effectiveness of adversary training for achieving counterfactual fairness. According to the table, we observe that the AUC scores of the baseline models are significantly higher than 0.5, which means that the attackers can easily discriminate user embeddings from sensitive features. In other words, for general recommendation models, no matter they are shallow or deep, the sensitive information of users can be learned from the data even such information is not explicitly used, thus leads to unfair recommendation results. Furthermore, the AUC scores of both SM and CM methods are around 0.5, i.e., it is hard for the attackers to distinguish the sensitive features from the filtered user embeddings. It indicates that both CM and SM implementations are effective for achieving counterfactual fairness in recommendations. Addtionally, from the observations of the AUC scores for SM and CM methods, we see that SM achieves lower AUC than CM in most cases, which shows that training separate filter functions for each combination of sensitive features can usually remove more sensitive information than the combination method.

Recommendation Performance. We compare the recommendation performance of the baseline models and the two fair methods. We show NDCG and Hit results of all the models on the MoiveLens and Insurance datasets in Table 3 and Table 4, respectively. We can see that both of the two fair methods can still achieve high recommendation quality. Although fair methods will suffer from a little sacrifice on recommendation performance to guarantee the fairness requirement, the recommendation performance is still very close to the original performance. This is acceptable as there is typically an inevitable trade-off between prediction accuracy and fairness [12, 44, 64]. The reason why there exists trade-off between fairness and recommendation performance is that the fair methods are aiming at filtering out the information of certain sensitive features from user embeddings, which will to some extent reduce the information contained in the embeddings, thus decreasing the recommendation performance.

Furthermore, comparing the recommendation performance of SM and CM, we can see that SM performs better on most of the metrics on the two datasets, especially for the case of single feature. We analyze the phenomenon as follows: for CM method, although we aim at learning one filter function for each sensitive



(a) Hit@5-G (b) Hit@5-A (c) Hit@5-O (d) AUC-G (e) AUC-A (f) AUC-O

Figure 6: Impact of the adversarial coefficient λ on (a)-(c) recommendation performance *w.r.t* Hit@5, and (d)-(f) classification quality of the attackers *w.r.t* AUC. In the subfigure titles, G, A and O represent the sensitive features *gender*, *age* and *occupation*, respectively. The reported results are from the Biased-MF model on the MovieLens dataset.

feature and wish that the filter function f_k only filters out the information of the k-th sensitive feature, we actually train the combination of all the filter functions together. Such learning process will force the filter functions to also remove the information that they should not remove. For example, if one user chooses a mask $\mathbf{Q}_u = \{gender, age\}$, and we use CM to generate user embedding as $\mathbf{r}_u^* = \frac{1}{2}(f_g(\mathbf{r}_u) + f_a(\mathbf{r}_u))$. When we train the filter functions to require that discriminators cannot classify *gender* and *age* from $\mathbf{r}_u^*$, f_g may be forced to also remove some information of *age* to satisfy the requirement, and it is the same for f_a. However, for SM method, we train one filter for each combination of sensitive features, thus the filter f_g will only filter out the information of *gender* as it has no contact with other features during the learning process. Therefore, comparing the recommendation performance of f_g trained by SM and CM method, we will find that SM filter performs better as it keeps more information for making recommendations, and such performance is even more significant for the evaluation of the single feature case.

However, SM method also has drawbacks as it will be infeasible when the potential number of combinations of the sensitive features is a lot. During the experiments, we found that SM usually needs more epochs to converge than CM, which will take more time for the training process. It is reasonable since SM method has more filter functions than CM, which needs extra epochs to make all filter functions be sampled and trained sufficiently.

Therefore, we suggest using SM to achieve fair models when the number of sensitive features is small to keep better recommendation performance while use CM to handle the situations where there are too many combinations of sensitive features.

5.6 Ablation Study

We study the influence of the adversarial coefficient λ on the recommendation performance and fairness in this section. As discussed before, λ controls the trade-off between recommendation quality and fairness. Theoretically, the larger λ is, the greater the influence of the discriminator loss will be in the whole loss, which means that we have a stricter demand for fairness and may have to scarify more recommendation performance to meet the requirement. And when $\lambda \to \infty$, there is a trivial solution that the filter functions always output a constant, resulting in a fair result but losing all the information for making accurate recommendations. To verify the influence of λ, we draw the change of AUC scores and the recommendation performance with the change of λ in Figure 6. Since similar trend is observed for other recommendation models, datasets, metrics, and the combinations of sensitive features, we plot the results of Biased-MF on MoiveLens under three single feature cases to keep the figure clarity. We can see that the experimental results are consistent with our analysis above. i.e., with the increase of λ, the recommendation performance has been declining, while the AUC score is getting lower and lower, which means that the system is getting fairer.

6 CONCLUSION

In this paper, we study the fairness problem for users in recommender systems. To better assess fairness in recommendation, we adopt causal-based fairness notions to reason about the causal relations between the protected features and the predicted results, instead of merely considering the traditional association-based fairness notions. We also consider personalized fairness which allows different users to have different fairness demands. Technically, to implement individual-level fairness for users, we approach counterfactual fairness for recommendation. We propose to generate feature-independent user embeddings to satisfy the counterfactual fairness requirements in recommendation, and we introduce an adversary learning method to learn such feature-independent user embeddings. Experiments on two real-world datasets with several shallow or deep recommendation algorithms show that our method is able to generate counterfactually fair recommendations for users with a desirable recommendation performance.

This work is one of our first steps towards personalized fairness under counterfactual notions in recommendation systems, and there is much room for future improvements. Except for the recommendation scenario that we considered in this work, we believe personalized fairness is also important for other intelligent systems such as search engines, social networks, language modeling and image processing, which we will consider in the future.

ACKNOWLEDGEMENT

We thank the reviewers for the reviews and suggestions. This work was supported in part by NSF IIS-1910154 and IIS-2007907. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

REFERENCES

[1] Himan Abdollahpouri and Robin Burke. 2019. Multi-stakeholder recommendation and its connection to multi-sided fairness. *arXiv preprint arXiv:1907.13158* (2019).
[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling popularity bias in learning-to-rank recommendation. In *RecSys*.
[3] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. *arXiv preprint arXiv:1901.07555* (2019).

[4] Gediminas Adomavicius and YoungOk Kwon. 2011. Improving aggregate recommendation diversity using ranking-based techniques. *TKDE* (2011).

[5] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 217–253.

[6] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*. PMLR, 60–69.

[7] Mario Arduini, Lorenzo Noci, Federico Pirovano, Ce Zhang, Yash Raj Shrestha, and Bibek Paudel. 2020. Adversarial Learning for Debiasing Knowledge Graph Embeddings. *International Workshop on Mining and Learning with Graphs* (2020).

[8] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.

[9] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *KDD*. 2212–2220.

[10] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *2017 Workshop on Fairness, Accountability & Transparency in Machine Learning* (2017).

[11] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*. 405–414.

[12] Avishek Bose and William Hamilton. 2019. Compositional fairness constraints for graph embeddings. In *ICML*.

[13] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.

[14] Robin Burke. 2017. Multisided fairness for recommendation. *arXiv preprint arXiv:1707.00093* (2017).

[15] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. 2017. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840* (2017).

[16] Hanxiong Chen, Shaoyun Shi, Yunqi Li, and Yongfeng Zhang. 2021. Neural Collaborative Reasoning. *WWW* (2021).

[17] Heng-Tze Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[18] Jaewoong Cho, Gyeongjo Hwang, and Changho Suh. 2020. A fair classifier using mutual information. In *ISIT*. IEEE, 2521–2526.

[19] Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. 2019. Two-player games for efficient non-convex constrained optimization. In *ALT*. 300–332.

[20] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2020. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems* (2020).

[21] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*. 214–226.

[22] Michael D Ekstrand, John T Riedl, and Joseph A Konstan. 2011. *Collaborative filtering recommender systems*. Now Publishers Inc.

[23] Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. *EMNLP* (2018).

[24] Zuohui Fu, Yikun Xian, Ruoyuan Gao, Jieyu Zhao, Qiaoying Huang, Yingqiang Ge, Shuyuan Xu, Shijie Geng, Chirag Shah, Yongfeng Zhang, and Gerard de Melo. 2020. Fairness-Aware Explainable Recommendation over Knowledge Graphs. *SIGIR* (2020).

[25] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, and Yongfeng Zhang. 2021. Towards Long-term Fairness in Recommendation. *WSDM* (2021).

[26] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander. 2016. Satisfying real-world goals with dataset constraints. In *NeurIPS*. 2415–2423.

[27] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. 1992. Using collaborative filtering to weave an information tapestry. *Commun. ACM* (1992).

[28] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *Neural Information Processing Systems* (2014).

[29] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *NeurIPS*. 3315–3323.

[30] Wen Huan, Yongkai Wu, Lu Zhang, and Xintao Wu. 2020. Fairness through equality of effort. In *Companion Proceedings of the Web Conference 2020*. 743–751.

[31] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. In *ECML*.

[32] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2014. Correcting Popularity Bias by Enhancing Recommendation Neutrality. In *RecSys*.

[33] Aria Khademi, Sanghack Lee, David Foley, and Vasant Honavar. 2019. Fairness in algorithmic decision making: An excursion through the lens of causality. In *The World Wide Web Conference*. 2907–2914.

[34] Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. 2017. Avoiding discrimination through causal reasoning. *arXiv preprint arXiv:1706.02744* (2017).

[35] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[36] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[37] Matt Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *NeurIPS*. 4069–4079.

[38] Jurek Leonhardt, Avishek Anand, and Megha Khosla. 2018. User Fairness in Recommender Systems. In *Companion Proceedings WWW*. 101–102.

[39] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. *WWW* (2021).

[40] Xiao Lin, Min Zhang, Yongfeng Zhang, Zhaoquan Gu, Yiqun Liu, and Shaoping Ma. 2017. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 107–115.

[41] P. Lops, M. De Gemmis, and G. Semeraro. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook* (2011).

[42] K. Makhlouf, S. Zhioua, and C. Palamidessi. 2020. Survey on Causal-based Machine Learning Fairness Notions. *arXiv preprint arXiv:2010.09553* (2020).

[43] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *CIKM*.

[44] A. K. Menon and R. C. Williamson. 2018. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*. PMLR.

[45] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[46] Harikrishna Narasimhan. 2018. Learning with complex loss functions and constraints. In *International Conference on Artificial Intelligence and Statistics*.

[47] Harikrishna Narasimhan, Andrew Cotter, Maya R Gupta, and Serena Wang. 2020. Pairwise Fairness for Ranking and Regression.. In *AAAI*. 5248–5255.

[48] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. 2020. FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms. In *WWW*.

[49] Judea Pearl. 2009. *Causality*. Cambridge university press.

[50] Judea Pearl. 2013. Direct and indirect effects. *arXiv:1301.2300* (2013).

[51] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

[52] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. 2009. Measuring discrimination in socially-sensitive decision records. In *SDM*. SIAM, 581–592.

[53] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *KDD*. 560–568.

[54] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *UAI* (2012).

[55] Shaoyun Shi, Hanxiong Chen, Weizhi Ma, Jiaxin Mao, Min Zhang, and Yongfeng Zhang. 2020. Neural Logic Reasoning. In *CIKM*. 1365–1374.

[56] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *KDD*. 2219–2228.

[57] Ashudeep Singh and Thorsten Joachims. 2019. Policy learning for fairness in ranking. In *Advances in Neural Information Processing Systems*. 5426–5436.

[58] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. 2019. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *ICCV*. 5310–5319.

[59] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. 2017. Learning non-discriminatory predictors. *arXiv:1702.06081* (2017).

[60] Yongkai Wu, Lu Zhang, Xintao Wu, and Hanghang Tong. 2019. Pc-fairness: A unified framework for measuring causality-based fairness. *NeurIPS* (2019).

[61] Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. *NeurIPS* (2017).

[62] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems.. In *IJCAI*.

[63] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *Advances in Neural Information Processing Systems*. 2921–2930.

[64] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*. 1171–1180.

[65] Meike Zehlike and Carlos Castillo. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *WWW*. 2849–2855.

[66] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *ICML*. 325–333.

[67] Junzhe Zhang and Elias Bareinboim. 2018. Equality of opportunity in classification: A causal approach. In *NeurIPS*. 3675–3685.

[68] Junzhe Zhang and Elias Bareinboim. 2018. Fairness in decision-making—the causal explanation formula. In *AAAI*, Vol. 32.

[69] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *CIKM*. 1449–1458.

[70] Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends in Information Retrieval* (2020).

[71] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*. 83–92.

[72] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. *CIKM* (2020).