SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2022, Vol. 65 222102:1–222102:15
https://doi.org/10.1007/s11432-020-3404-y

FairCF: fairness-aware collaborative filtering

Pengyang SHAO[1], Le WU[1,2,3*], Lei CHEN[1], Kun ZHANG[1,2] & Meng WANG[1,2,3*]

[1]*School of Computer Science and Information Engineering, Hefei University of Technology, Hefei* 230601*, China;*
[2]*Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei* 230601*, China;*
[3]*Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei* 230088*, China*

Received 26 December 2020/Revised 8 April 2021/Accepted 30 August 2021/Published online 17 November 2022

Abstract Collaborative filtering (CF) techniques learn user and item embeddings from user-item interaction behaviors, and are commonly used in recommendation systems to help users find potentially desirable items. Most CF models optimize recommendation accuracy; however, they may lead to unwanted biases for particular demographic groups. Thus, we focus on learning fair representations of CF-based recommendations. We formulate this problem as an optimization task with two competing goals: embedding representations better meet accuracy requirements of recommendations, and simultaneously obfuscate information hidden in the embedding space, which is related to the users' sensitive attributes for fairness. Here, the intuitive idea is to use fair representation learning from machine learning to train a classifier with a sensitive attribute predictor from the user side to satisfy the fairness goal. However, such fair machine learning models assume entity independence, which differs greatly from CF because users and items are correlated collaboratively via user-item behaviors. Therefore, sensitive user information can be exposed from the users' preferred items. Consequently, defining only fairness constraints on users cannot achieve fairness in recommendation systems. In this paper, we propose FairCF framework for fairness-aware collaborative filtering. In particular, we first define fairness constraints in a fair embedding space, where both a user classifier and an item classifier are employed to fit the fairness constraints. We then design an item classifier without item sensitive labels. The proposed framework can be trained in an end-to-end manner under most embedding based CF models. Extensive experiments conducted on three datasets (MovieLens-100K, MovieLens-1M, and Lastfm-360K) clearly demonstrate the superiority of the proposed FairCF framework relative to various fairness metrics (i.e., performance of newly-trained classifiers) than other state-of-the-art fairness-aware CF models with less than 4% accuracy reduction.

Keywords recommendation systems, fairness, adversarial learning

Citation Shao P Y, Wu L, Chen L, et al. FairCF: fairness-aware collaborative filtering. Sci China Inf Sci, 2022, 65(12): 222102, https://doi.org/10.1007/s11432-020-3404-y

1 Introduction

Recommendation systems help users identify potential interests in various items and are used widely [1–4]. Collaborative filtering (CF) is a popular technique in recommendation systems due to its relatively high performance and easy-to-collect user-item interaction data [5–9]. CF models assume that users and items are correlated collaboratively from their behavior data, and state-of-the-art CF models focus on learning accurate user and item embeddings from the user-item interaction data. Then, users' preferences for items can be predicted directly by comparing the similarity between the learned user and item embeddings. For example, classical latent factor models learn user and item embeddings using matrix factorization techniques [5, 7, 10]. Recently, studies have argued that users and items naturally form a user-item bipartite graph, and neural graph models have exhibited superior performance compared to classical latent factor models [3, 8].

Most CF-based recommendation models focus on recommendation accuracy; however, some researchers have argued that user-centric recommendation may inherit biases in training data and lead to unfairness issues, thereby potentially discriminating against a specific population. For example, a job recommendation platform, Xing, was found that more qualified female candidates are ranked lower than less qualified

* Corresponding author (email: lewu.ustc@gmail.com, eric.mengwang@gmail.com)

male candidates[1)] [11]. News recommendation can easily capture biases related to gender and leads to recommending biased news for users [12]. Ad recommendation shows obvious racial biases among users with similar interests [13]. Rather than facilitating unfair recommendations that favor a particular demographic group, ideally, we would like to construct a recommendation model that provides accurate recommendation results and is not discriminatory to any sensitive user groups.

Ensuring fairness in user-related machine learning has received increasing attention in recent years. Most studies in this area have focused on fair user classification tasks [14–16]. Given a sensitive attribute, users are binned into different subgroups based on the detailed sensitive attribute value. These models attempt to optimize two competing goals simultaneously, i.e., maximizing classification accuracy and minimizing the classification prediction differences of different subgroups to achieve fairness [16,17]. Various models have been proposed to add fairness based regularization terms to these classification tasks [18]. Recently, with the huge success of representation learning, many approaches have turned to learning fair representations to satisfy two goals, i.e., the learned representations are predictive for downstream tasks, and no sensitive information is encoded in the representations [19–22]. Specifically, adversarial training is widely used to learn fair representations, and an additional sensitive classifier is trained to predict the sensitive attribute [20,21]. The encoder and sensitive attribute classifier play a minimax game to match the conditional distribution of the representations of each subgroup to satisfy fairness requirements [22]. Similarly, some studies have attempted to implement debias in recommendations. For example, previous studies added carefully designed fairness regularization terms of subgroup users in matrix factorization-based CF models [23] or relied on learning adversarial fair embeddings of users to match embedding distributions of different subgroups [24]. Such fair recommendation models enhance recommendation fairness and maintain high recommendation accuracy.

Despite the success of these fair recommendation models, we argue that current fair CF models are still not satisfactory. Most fair recommendation models inherit fairness techniques from fair classification in machine learning, and they assume that each instance is independent in the modeling process. In CF systems, there are two kinds of entities, i.e., user and item entities. Users and items are not independent but are related collaboratively. This collaborative correlation is also reflected in the user and item embedding learning process because users and items are mapped in the same low latent space for recommendation prediction. Thus, if we borrow fair classification techniques with user side fairness consideration, the users' sensitive attributes will still be exposed via their item behaviors. Consider a recommendation system that recommends different types of shoes. In this system, we could not acquire gender information from the user side. However, if we find that a given user has clicked on high-heeled shoes many times, we can infer that this user is probably female. Even though we can apply fair representation learning techniques to ensure that the users' representation are fair without any sensitive attribute information, the items the users interact with will still expose their sensitive attributes, which results in unfairness issues.

Therefore, in this paper, we investigate how fair representations can be learned for CF-based recommendation systems. Here, an intuitive concept is to play a two-player minimax game between the CF algorithm and an additional sensitive attribute classifier. Specifically, the classifier attempts to infer labels of sensitive user information from user embeddings, and the embedding-based CF module prevents the classifier from predicting the sensitive attribute. Under such an adversarial training procedure, the representation distribution differences among different user subgroups are reduced [22,25]. However, users and items are correlated collaboratively; thus, sensitive user information can be inferred from the item rated by the user. To consider the collaborative correlation between users and items, we define fairness constraints on item embeddings and propose an item adversarial module to alleviate unfairness caused by this user-item correlation. The item adversarial module is difficult to implement because items do not have sensitive label information. To address this issue, we first demonstrate how pseudo labels of items can be assigned based on the given sensitive labels of the users. We then design a FairCF framework that encourages fairness on both the user and item sides. FairCF involves three loss terms, i.e., an accuracy-based loss term that measures recommendation accuracy and two classification-based loss that attempt to eliminate unfairness exposed by user-sensitive labels and pseudo labels of items. The proposed FairCF framework is flexible and can be applied to state-of-the-art embedding-based CF models. Extensive experiments were conducted on three real-world datasets to demonstrate the effectiveness of the proposed framework. In summary, our primary contribution lies in discovering the correlations of users and items

1) https://www.xing.com/.

for fairness modeling, proposing a simple solution to tackle unfairness caused by the correlations, and validating the proposed framework through extensive experimental results.

2 Related work

2.1 Recommendation systems

Recommendation systems have been widely used to help users find potential items of interest [1, 26, 27]. Typically, recommendation systems involve two kinds of entities, i.e., a user set U ($|U| = M$) and an item set V ($|V| = N$). Interactions between users and items can be represented as an interaction matrix $\boldsymbol{R} = \{r_{uv}\}_{M\times N}$, where r_{uv} denotes the interaction between user u and item v. Specifically, if user u has rated item v, the interaction r_{uv} equals the true rating, forming observed triplets (u, v, r_{uv}). If user u has not rated item v, then $r_{uv} = 0$. Learning high-quality user and item embeddings is the foundation of modern recommendation systems [7, 8, 28, 29]. Let $\boldsymbol{E} = [\boldsymbol{E}_U, \boldsymbol{E}_V] = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_u, \ldots, \boldsymbol{e}_M, \ldots, \boldsymbol{e}_v, \ldots, \boldsymbol{e}_{M+N}] \in \mathbb{R}^{(M+N)\times D}$ denote the learned embedding space, where $\boldsymbol{E}_U = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_M]$ represents user embeddings, and $\boldsymbol{E}_V = [\boldsymbol{e}_{M+1}, \ldots, \boldsymbol{e}_{M+N}]$ represents item embeddings. Here, D denotes the latent factor dimension, and $\boldsymbol{e}_u$ and $\boldsymbol{e}_v$ denote user u's and item v's corresponding embeddings in $\boldsymbol{E}$, respectively. Based on the learned embeddings, the predicted preference $\hat{r}_{uv}$ of user u for item v is calculated as follows:

$$\hat{r}_{uv} = \boldsymbol{e}_u^{\mathrm{T}} \boldsymbol{e}_v + b_u + b_v + \mu, \tag{1}$$

where b_u and b_v represent user biases and item biases, respectively, and μ represents the global average rating. Here, $\boldsymbol{e}_u^{\mathrm{T}} \boldsymbol{e}_v$ denotes the inner product between $\boldsymbol{e}_u$ and $\boldsymbol{e}_v$.

Classical latent factor-based models apply matrix factorization to learn the user and item embeddings [5, 6]. Users and items naturally form a user-item bipartite graph $G = \langle U \cup V, \boldsymbol{R} \rangle$ with interactions $\boldsymbol{R}$; thus, neural graph-based models have been proposed recently to learn user and item embeddings [3, 8]. The key idea of neural graph-based models is to update each user's (item's) higher order embedding iteratively by aggregating neighborhood embeddings in the previous layer. Therefore, the final user (item) embedding contains the user's (item's) subgraph structure in the user-item bipartite graph. These neural graph-based models exhibit better performance than matrix factorization models because they can alleviate the data sparsity issue in CF by injecting CF signals for representation learning.

2.2 Fairness in machine learning

Studies have found that machine learning models inherit biases in the training data and exhibit discrimination [15, 30]. In addition, different fairness definitions have been proposed. Individual fairness requires that similar individuals should be treated similarly [11, 31]. Counterfactual fairness ensures the same treatment in the factual world or a counterfactual world [32]. Among these fairness definitions, group fairness, which has been widely studied [33–35], ensures the same treatment for different groups. In this paper, we focus on fair representation learning for group fairness due to the generality and recent rapid developments in representation learning. Fair representation learning performs feature learning to facilitate both downstream tasks and group-based fairness requirements on the learned representations [19–22]. A regularization-based fair representation framework has been proposed previously, where the representations encode data and obey the statistical parity principle [19]. Based on adversarial learning from generative adversarial nets [25], an adversarial method has also been proposed to learn fair representations [20]. Group fairness definitions were theoretically proven to connect to adversarial training objectives [21]. The connections between fair representations and fair downstream tasks were proven both theoretically and experimentally [22]. We follow this line of adversarial fair representation learning and focus on applying it to CF models.

2.3 Fairness in recommendation systems

Recommendation is one of the most widely used user-centric applications; thus, fairness issues have attracted attention in recommendation systems [23, 24, 36–39]. In recommendation systems, vulnerable user groups can be treated unequally [40, 41]. According to different principles of group fairness in recommendation, researchers have defined different fairness goals, and a regularization-based method has been proposed to reduce discrepancies between disadvantaged and advantaged users [23]. To isolate and

remove sensitive information from the latent factor embedding space, a fairness-aware tensor-based recommendation model that involves a sensitive information regularizer has been proposed previously [36]. Rather than directly removing unfairness in the model learning process, a previous study also applied model reranking in search and recommendation systems [37]. In addition, a fine-tuning approach on neural CF models was proposed to mitigate gender bias in sensitive item recommendation [38], and a fairness-aware news recommendation approach with decomposed adversarial learning has also been proposed [12]. Different sensitive attributes are correlated; thus, a method has been proposed to remove compositional unfairness using a composition of filters with adversarial training techniques [24]. Considering the graph structure of recommendation systems, a graph-based technique has also been proposed to ensure fairness [39]. Our work differs from these existing models because we argue that users and items are correlated collaboratively in recommendations systems, where a user's sensitive attributes can be classified easily according to the items that the user interacts with, and we consider how to ensure fairness in CF while considering the user-item correlation.

3 Proposed framework

Here, we introduce the proposed fairness-aware collaborative filtering (FairCF) framework. We first discuss preliminaries, followed by a detailed discussion of the overall architecture of FairCF. Then, we describe the training procedures of the proposed FairCF.

3.1 Preliminaries

Most CF models employ embedding learning techniques to represent users and items in a latent space. Here, a user's predicted preferences for an item can be calculated as the inner product between the user and item representations. Following the representation learning framework for CF, FairCF has two goals, i.e., maintaining recommendation accuracy to provide high-quality recommendations for users, and simultaneously improving fairness to ensure that users receive equal treatments. The accuracy goal is similar to previous CF models, where an accurate embedding matrix $\boldsymbol{E} = [\boldsymbol{E}_U, \boldsymbol{E}_V]$ is learned from user's sparse interaction records $\boldsymbol{R}$. A user's sensitive information can be exposed from the user's interactive items. For any sensitive attribute feature, e.g., gender, we denote the sensitive attribute values of all users as $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_u, \ldots, \boldsymbol{s}_M] \in \mathbb{R}^M$, where $\boldsymbol{s}_u$ represents user u's sensitive attribute value. The fairness goal focuses on modeling and removing unfairness caused by the sensitive attributes in the embedding space $\boldsymbol{E}$.

An adversarial framework has been proposed previously to satisfy the fairness goal by removing sensitive information from the embedding space [24]. This adversarial framework requires a classical embedding-based model $\mathcal{E}$ in CF that outputs embedding matrix $\boldsymbol{E}$, a filter module $\mathcal{F}$ that takes the original user embeddings $\boldsymbol{E}_U = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_u, \ldots, \boldsymbol{e}_M]$ to filter user embeddings $\boldsymbol{F}_U = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_u, \ldots, \boldsymbol{f}_M]$, and a classifier $\mathcal{D}_1$ that performs adversarial training to realize fairness. Specifically, the original embedding module $\mathcal{E}$ can take any classical embedding-based CF models, e.g., PMF [5] and LR-GCCF [8]. The $\mathcal{F}$ filter removes sensitive information that is correlated with user embeddings and outputs a filtered user embedding matrix $\boldsymbol{F}_U$ that encourages both recommendation accuracy and recommendation fairness.

When the original embeddings $\boldsymbol{E}_U$ are transformed to the filtered embeddings $\boldsymbol{F}_U$, the predicted rating $\hat{r}_{uv}$ of user u for item v can be calculated as follows:

$$\hat{r}_{uv} = \boldsymbol{f}_u^{\mathrm{T}} \boldsymbol{e}_v + b_u + b_v + \mu. \tag{2}$$

Given the above predicted ratings, any accuracy-based loss can be used, e.g., rating point-wise based loss [5] and pair-wise based loss [6]. Without loss of generality, we use the point-wise based squared loss, which is denoted $L_{\mathcal{E}}$:

$$L_{\mathcal{E}}(\boldsymbol{F}_U, \boldsymbol{E}_V) = \frac{1}{T} \sum_{(u,v,r_{uv}) \in (U,V,\boldsymbol{R})} (r_{uv} - \hat{r}_{uv})^2, \tag{3}$$

where u, v, and r_{uv} (i.e., the user u, item v, and corresponding ground truth rating r_{uv}, respectively) are sampled from training data $(U, V, \boldsymbol{R})$. In addition, T is the number of interactions in the training data.

The fairness goal is achieved in the adversarial training step. The classifier $\mathcal{D}_1$ and filter module $\mathcal{F}$ play a minimax game. Here, $\mathcal{F}$ attempts to avoid exposing any sensitive information, and $\mathcal{D}_1$ attempts to correctly infer the sensitive attribute. Via adversarial training of these two modules, studies have

shown that the filtered embedding distribution of different subgroups tends to be similar with some conditions [24, 25]. Next, we introduce the adversarial steps. Here, let $\hat{\boldsymbol{s}}_u$ be the classification result of inferring user u's sensitive attribute:

$$\hat{\boldsymbol{s}}_u = \mathcal{D}_1(\boldsymbol{f}_u). \tag{4}$$

Let K denote the number of sensitive attribute categories. The sensitive attribute category of each user u can be represented via one-hot coding as follows: $\boldsymbol{s}_u = [s_u^1, \ldots, s_u^k, \ldots, s_u^K]$. Note that we optimize the classifier $\mathcal{D}_1$ using a cross-entropy loss function as follows:

$$L_{\mathcal{D}_1}(\boldsymbol{F}_U, \boldsymbol{S}) = -\frac{1}{M}\sum_{u=1}^{M}\sum_{k=1}^{K} s_u^k \log \hat{s}_u^k. \tag{5}$$

Given the recommendation module $\mathcal{E}$, the filter module $\mathcal{F}$, and the sensitive attribute classification module $\mathcal{D}_1$, the learning procedures can be represented as playing a minimax game. We follow the form of minimax game presented in the literature [25]. Then, the procedures can be formulated as follows:

$$\min_{\mathcal{E},\mathcal{F}} \max_{\mathcal{D}_1} L_{\mathcal{E}}(\boldsymbol{F}_U, \boldsymbol{E}_V) - \beta L_{\mathcal{D}_1}(\boldsymbol{F}_U, \boldsymbol{S}), \tag{6}$$

where the first term is the accuracy goal with recommendation accuracy loss, and the second term models fairness. Here, β is a balancing parameter between fairness and accuracy. The optimal solution of the minimax game is a balancing point between accuracy and user embedding fairness. When the optimal solution is acquired, the recommendation system maintains recommendation accuracy and simultaneously eliminates unfairness from the user embeddings. If $\beta = 0$, Eq. (6) degenerates to classical recommendation accuracy-based approaches. If $\beta \to \infty$, the results are meaningless because Eq. (6) only considers the fairness issues, i.e., recommendation accuracy is not considered.

Before introducing the proposed framework, we demonstrate below why the above approach cannot thoroughly remove sensitive information to ensure fairness. Considering that users and items are correlated in the embedding space, items also represent risks relative to exposing sensitive user information. For example, on most movie platforms, women prefer romantic movies while men prefer action movies. By projecting users and items into the same embedding space with any classical CF model, it is easy to find that action movies are closer to men in the embedding space. Therefore, we could infer a user's gender based on the movies they watched in the item embedding space. In other words, due to the collaborative effect between users and items, sensitive user information is exposed in the item embedding matrix $\boldsymbol{E}_V$. Items do not have any sensitive label information on the user side; thus, filtering item embeddings at the same time is a challenge.

3.2 Overall architecture of FairCF

To eliminate sensitive information from item embeddings $\boldsymbol{E}_V$, we can use an item classifier $\mathcal{D}_2$ and build an adversarial loss to ensure that sensitive user information is not exposed from item embeddings. Thus, item embeddings are also mapped to the filtered space. Naturally, we must map the original embedding matrix $\boldsymbol{E} = [\boldsymbol{E}_U, \boldsymbol{E}_V]$ to a filtered space $\boldsymbol{F} = [\boldsymbol{F}_U, \boldsymbol{F}_V]$. Specifically, the filtered item embeddings $\boldsymbol{F}_V$ comprise $[\boldsymbol{f}_{M+1}, \ldots, \boldsymbol{f}_v, \ldots, \boldsymbol{f}_{M+N}]$, where $\boldsymbol{f}_v$ denotes item v's filtered item embedding.

A fair embedding space must satisfy recommendation accuracy and embedding space fairness. As mentioned in Subsection 3.1, fair user embeddings can be achieved by adding a classifier $\mathcal{D}_1$ to infer sensitive attributes from the user embeddings via adversarial training. Similarly, we propose an item classifier $\mathcal{D}_2$ to predict the sensitive information hidden in item embeddings. Note that we discuss solving the problem of no item sensitive labels in the following subsection. Here, we assume that all items have sensitive labels $\boldsymbol{P}$ and introduce FairCF .

As shown in Figure 1, the filter module is applied to both the user and item embeddings learned using a classical CF model $\mathcal{E}$. Given the filtered embedding space, the predicted rating $\hat{r}_{uv}$ of user u for item v is calculated in the filtered embedding space as follows:

$$\hat{r}_{uv} = \boldsymbol{f}_u^{\mathrm{T}} \boldsymbol{f}_v + b_u + b_v + \mu. \tag{7}$$

Here, two adversarial modules are applied to eliminate sensitive information from the user and item embeddings. By taking the filtered embeddings as input, the user classifier attempts to predict the user's sensitive labels, and the item classifier attempts to predict the pseudo sensitive labels of the items.

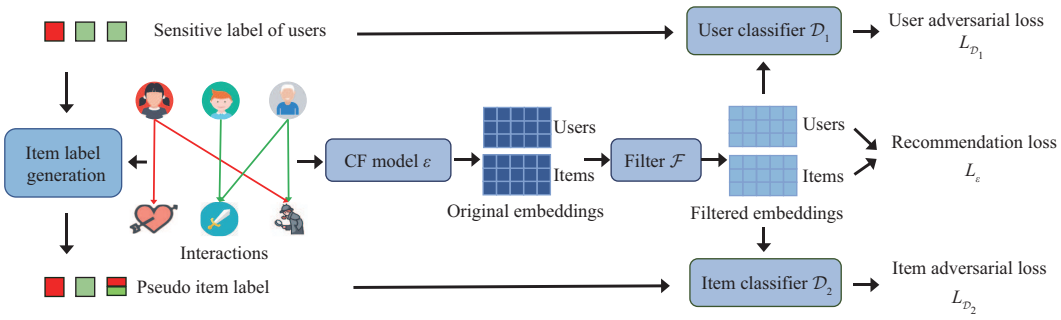

Figure 1 (Color online) Overall structure of our proposed FairCF.

Then, we define the overall loss function as follows:

$$L_{\text{all}} = L_{\mathcal{E}}(\boldsymbol{F}_U, \boldsymbol{F}_V) - \beta L_{\mathcal{D}_1}(\boldsymbol{F}_U, \boldsymbol{S}) - \gamma L_{\mathcal{D}_2}(\boldsymbol{F}_V, \boldsymbol{P}), \tag{8}$$

where the first term is recommendation accuracy loss, the second term represents the user's sensitive classification results, and the third term is the classification results from pseudo item sensitive labels. In addition, β and γ are balancing parameters that control the classification results. When γ equals zero, the classification results from pseudo item sensitive labels disappear. Similar to (6), here, we can optimize the overall optimization function using a minimax game as follows:

$$\min_{\mathcal{E},\mathcal{F}} \max_{\mathcal{D}_1,\mathcal{D}_2} L_{\mathcal{E}}(\boldsymbol{F}_U, \boldsymbol{F}_V) - \beta L_{\mathcal{D}_1}(\boldsymbol{F}_U, \boldsymbol{S}) - \gamma L_{\mathcal{D}_2}(\boldsymbol{F}_V, \boldsymbol{P}), \tag{9}$$

where $L_{\mathcal{D}_1}(\boldsymbol{F}_U, \boldsymbol{S})$ is realized by (5), and $L_{\mathcal{E}}(\boldsymbol{F}_U, \boldsymbol{F}_V)$ is employed to improve recommendation accuracy as follows:

$$L_{\mathcal{E}}(\boldsymbol{F}_U, \boldsymbol{F}_V) = \frac{1}{T} \sum_{(u,v,r_{uv})\in(U,V,\boldsymbol{R})} (r_{uv} - \hat{r}_{uv})^2. \tag{10}$$

In the following, we emphasize how pseudo item labels are assigned and discuss the design of the loss function $L_{\mathcal{D}_2}(\boldsymbol{F}_V, \boldsymbol{P})$.

3.2.1 *Pseudo item label calculation*

As discussed in Subsection 3.1, items are not labeled with sensitive attribute values. There is no effective guidance to eliminate sensitive information from the item side. However, users and items are correlated collaboratively; thus, by passing user sensitive information through the correlated structure, pseudo item labels are proposed to eliminate sensitive information from the item side. Here, we must select a deterministic method for pseudo item label prediction. Since we only have user-item behavior data, we predict the item pseudo labels from the observed user labels. Specifically, we consider the distribution "item pseudo label" as the percentage of linked users that have this particular sensitive attribute value. Here, p_v^k is the probability of the k-th sensitive attribute value of item v. Thus, we define the pseudo item sensitive label distribution as follows:

$$p_v^k = \frac{\sum_{u\in\mathcal{B}_v}^{s_u=k} 1}{\sum_{u\in\mathcal{B}_v} 1}, \tag{11}$$

where $\mathcal{B}_v$ is the user subset that connects to item v, i.e., the user subset that rated item v. In addition, $\sum_{u\in\mathcal{B}_v}^{s_u=k} 1$ is the number of users in $\mathcal{B}_v$ whose sensitive attribute values are k, and $\sum_{u\in\mathcal{B}_v} 1$ is the number of users in $\mathcal{B}_v$. Thus, we can calculate all pseudo item labels for all items, i.e., $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_v, \ldots, \boldsymbol{p}_N] \in \mathbb{R}^{K\times N}$.

We also evaluated other ways to calculate the pseudo item labels, e.g., by adding the rating values as the relative weight to replace interactions in (11). However, we found that the experimental results with different settings were similar. In addition, in real-world settings, most users would only show implicit interaction data rather than the exact rating values. In summary, the proposed pseudo item label calculation model can be applied to most recommendation scenarios with relatively high effectiveness.

3.2.2 *Item adversarial loss*

After defining the pseudo sensitive item labels $\boldsymbol{P}$, we require an item classifier $\mathcal{D}_2$ to eliminate sensitive information leakage. By taking item embeddings $\boldsymbol{F}_V$ from the filter network as input, the item classifier attempts to guess the pseudo item labels $\boldsymbol{P}$ as follows:

$$\hat{\boldsymbol{p}}_v = \mathcal{D}_2(\boldsymbol{f}_v), \tag{12}$$

where $\hat{\boldsymbol{p}}_v$ represents the predicted values of item v's pseudo sensitive labels. Note that the pseudo item label values are continuous in the range of 0 to 1; thus, we use the mean square error loss to optimize the item classifier $\mathcal{D}_2$ as follows:

$$L_{\mathcal{D}_2}(\boldsymbol{F}_V, \boldsymbol{P}) = \frac{1}{N}\sum_{v=1}^{N}(\hat{\boldsymbol{p}}_v - \boldsymbol{p}_v)^2 = \frac{1}{N}\sum_{v=1}^{N}(\mathcal{D}_2(\boldsymbol{f}_v) - \boldsymbol{p}_v)^2. \tag{13}$$

By comparing (6) and (9), it appears that the proposed FairCF only implements an additional adversarial loss. We argue that our main contribution lies in discovering and eliminating unfairness in the bipartite graph structure.

3.3 Training procedures of FairCF

Here, we describe the detailed training procedures of FairCF. The training procedures can be divided into two stages. The first stage involves pretraining, which is widely used in adversarial training [42]. Pretraining is required to reduce the convergence time and improve the quality of the generated embeddings. Specifically, by selecting a particular recommendation model, e.g., PMF [5] or LR-GCCF [8], we utilize (10) to pretrain the recommendation model followed by pretraining the user and item classifiers to ensure that they have some ability to infer sensitive labels. We then train the overall framework and stop model learning when both the performance of user and item classifiers no longer vary. The detailed training process is shown in Algorithm 1.

Algorithm 1 Detailed training procedure of FairCF

Require: Users U; items V; interactions $\boldsymbol{R}$; user sensitive labels $\boldsymbol{S}$.
Generate pseudo item labels $\boldsymbol{P}$ for all items (Eq. (11));
Randomly initialize all module $\mathcal{E}$, $\mathcal{F}$, $\mathcal{D}_1$, $\mathcal{D}_2$'s parameters $\Theta_{\mathcal{E}}$, $\Theta_{\mathcal{F}}$, $\Theta_{\mathcal{D}_1}$, $\Theta_{\mathcal{D}_2}$;
repeat
  Sample a batch of training data, including user-item pairs and corresponding interactions and sensitive labels $(u, i, r_{uv}, \boldsymbol{s}_u, \boldsymbol{p}_v)$;
  for Each pair of input $(u, i, r_{uv}, \boldsymbol{s}_u, \boldsymbol{p}_v)$ in the batch **do**
    Compute the recommendation loss $L_{\mathcal{E}}$ (Eq. (10));
    Optimize $\Theta_{\mathcal{E}}$, $\Theta_{\mathcal{F}}$ to minimize recommendation loss $L_{\mathcal{E}}$;
    Compute the user adversarial loss $L_{\mathcal{D}_1}$ (Eq. (5));
    Optimize $\Theta_{\mathcal{D}_1}$ to minimize user adversarial loss $L_{\mathcal{D}_1}$;
    Compute the item adversarial loss $L_{\mathcal{D}_2}$ (Eq. (13));
    Optimize $\Theta_{\mathcal{D}_2}$ to minimize item adversarial loss $L_{\mathcal{D}_2}$;
  end for
until pretraining stops.
repeat
  Sample a batch of training data, including user-item pairs and corresponding interactions and sensitive labels $(u, i, r_{uv}, \boldsymbol{s}_u, \boldsymbol{p}_v)$;
  Fix $\Theta_{\mathcal{D}_1}$, $\Theta_{\mathcal{D}_2}$ and compute the overall loss function L_{all} (Eq. (8));
  Optimize $\Theta_{\mathcal{E}}$, $\Theta_{\mathcal{F}}$ to minimize the overall loss L_{all};
  Fix $\Theta_{\mathcal{E}}$, $\Theta_{\mathcal{F}}$ and compute the user adversarial loss $L_{\mathcal{D}_1}$ (Eq. (5));
  Optimize $\Theta_{\mathcal{D}_1}$ to minimize user adversarial loss $L_{\mathcal{D}_1}$;
  Fix $\Theta_{\mathcal{E}}$, $\Theta_{\mathcal{F}}$ and compute the item adversarial loss $L_{\mathcal{D}_2}$ (Eq. (13));
  Optimize $\Theta_{\mathcal{D}_2}$ to minimize item adversarial loss $L_{\mathcal{D}_2}$;
until Convergence.

4 Experiments

Here, we first describe the experimental setup and then introduce our proposed recommendation fairness metric. Then, we present the experimental results and demonstrate that the proposed FairCF outperforms state-of-the-art algorithms. Finally, we present detailed analyses to facilitate further discussions.

Table 1 Statistics of the datasets

Datasets	Users	Items	Ratings	Density (%)
MovieLens-100K	943	1682	100000	6.30
MovieLens-1M	6040	3706	1000209	4.4684
Lastfm-360K	359347	292589	17559443	0.0167

4.1 Experimental setup

4.1.1 *Datasets*

In this evaluation, three datasets were considered to evaluate model performance.

• MovieLens-100K. This dataset is widely used for movie recommendation [43]. The dataset MovieLens-100K includes 100000 ratings from one to five rated by 943 users for 1682 movies. We randomly selected 70% of the triplets as training data and 10% for validation. The remaining 20% of the triplets were used as test data. The user demographic data include gender, age, and occupation.

• MovieLens-1M. This dataset includes movie recommendation provided by MovieLens users [43]. As shown in Table 1, the dataset comprises nearly one million ratings from one to five rated by 6040 users on approximately 4000 movies. Here, we randomly selected 70% of the triplets as training data and 10% for validation. The remaining 20% of the triplets were used as test data. The MovieLens-1M dataset includes user demographic data (age, zip code, gender, and occupation) and movie metadata (title and genres).

• Lastfm-360K. This dataset contains triplets (user, artist, play times) collected from the Last.fm API [44]. On top of interactive information, it comprises nearly 360000 users and approximately 290000 artists. Note that the Lastfm-360K dataset does not record ratings directly; thus, we change times into ratings using a log, followed by mapping the ratings into integers from 1 to 5. We split the triplets at a ratio of 7:1:2. Detailed statistics for this dataset are shown in Table 1. The Lastfm-360K dataset includes user demographic data (gender, age, country, and signup date).

4.1.2 *Baselines*

We compare our proposed FairCF with the following baselines:

• PMF. This is a classical CF model in the recommendation field [5].

• LR-GCCF. This is a graph-based recommendation model that treats the user-item interaction as a bipartite graph and simplifies the graph convolution aggregation operation [8]. Note that we modified this model that was originally used in implicit feedback to explicit feedback. Specifically, for the MovieLens-1M dataset, we replaced the concatenation operation with the addition operation when combining different layers of embeddings. In the Lastfm-360K dataset, we further utilized a layer-attention module to learn different weights when adding different user (item) embeddings from different layers to obtain the best accuracy.

• NIPS_Non and NIPS_Pop. This is a regularization-based method to reduce the impact of gender on recommendation results [23]. This approach considers different fairness regularization terms for different forms of unfairness. Specifically, we selected nonparity and population parity regularization, which are denoted NIPS_Non and NIPS_Pop, respectively. Note that these models are defined on PMF; thus, we only compared NIPS_Non and NIPS_Pop to PMF-based frameworks.

• ICML_2019. This is a state-of-the-art adversarial learning-based method to achieve fairness constraints [24]. Note that this baseline was introduced in Subsection 3.1.

• FairCF_NIC. This is a simplified version of the proposed framework implemented to investigate the effectiveness of the item classifier $\mathcal{D}_2$ in FairCF . This simplified version of FairCF has no item classifier $\mathcal{D}_2$, i.e., $\gamma = 0$ in (8).

In summary, PMF and LR-GCCF are basic CF models that do not consider fairness. To explicitly introduce fairness constraints, NIPS_Non and NIPS_Pop remove unfairness by adding fairness regularization terms. In addition, ICML_2019 attempts to eliminate sensitive information for user embeddings using adversarial training. FairCF_NIC is a simplified version of the proposed FairCF to validate its effectiveness.

Table 2 Hyperparameters of FairCF

Datasets	Batchsize	PMF		LR-GCCF		Times for	Times for	Times for
		β	γ	β	γ	training $\mathcal{E}$, $\mathcal{F}$	training $\mathcal{D}_1$	training $\mathcal{D}_2$
MovieLens-100K	32768	10	20	10	20	1	10	10
MovieLens-1M	8192	10	20	10	10	1	10	20
Lastfm-360K	1048576	5	10	40	80	1	20	20

4.1.3 *Implementation details*

Our experiments were implemented on pytorch-1.6.0. For hyperparameters, we set the matrix dimension $D = 64$ and initialize the embedding matrix $\boldsymbol{E}$ with the normal distribution $\mathcal{N}(0, 0.01^2)$. In addition, Adam was employed as the optimization method with a learning rate of 0.005. The coefficient of the regularization term was 0.001. We set different values for balancing parameters β and γ on all three datasets (MovieLens-100K, MovieLens-1M, and Lastfm-360K). We set different values for balancing parameters β and γ, and times of training $\mathcal{E}$, $\mathcal{F}$, $\mathcal{D}_1$, $\mathcal{D}_2$ modules during one epoch in FairCF varies on different datasets. These hyperparameters are shown in Table 2. The filter module $\mathcal{F}$ and two classifiers $\mathcal{D}_1$, $\mathcal{D}_2$ were implemented using multilayer perceptrons (MLP) with two, four, and four layers, respectively. Here, the filter module $\mathcal{F}$ served as a mapping function from the original embedding space to the target space without changing the embedding size.

For the sensitive attribute, we followed [21–23] and selected gender as the sensitive attribute because gender is an easy-to-collect attribute that represents information sensitivity in the real world.

4.2 Evaluation metrics

We evaluated the recommendation accuracy according to the root mean square error (RMSE) metric, which metric measures the differences between ground truth ratings and the predicted ratings. Note that smaller RMSE values indicate better recommendation accuracy.

Due to a lack of unified fairness metrics in the recommendation field, the performance acquired on a newly trained classifier is typically used to evaluate the fairness of user representations [12,24]. Here, we split users at a ratio of 8:2 and trained the classifier on data from 80% of the users. We evaluated the classifier performance of the remaining 20% of the users. As mentioned previously, we selected a binary sensitive attribute (i.e., gender): therefore, we used the area under the curve (AUC) metric to represent how much gender information remained in the user embeddings. Note that a smaller AUC value indicates better performance in terms of user embedding fairness.

In addition, we also considered a widely adopted group fairness metric of statistical parity [23] to measure the differences in the predicted ratings of different groups, i.e., $1/N \sum_{j=1}^{N} ||E_{u\in U_0}[\hat{r}_{uv}] - E_{u\in U_1}[\hat{r}_{uv}]||$, where U_0 and U_1 denote different user groups divided by the binary sensitive attribute (gender). The recommendation task is a ranking-oriented task; thus, the goal is to predict the top ranked items as accurately as possible. Therefore, in addition to measuring group fairness based on all predicted ratings, we also modified the statistical parity onto the top-K ranked items as follows:

$$\text{Fairness@}K = \frac{\sum_{j=1}^{N} |\frac{1}{|U_0|} \sum_{i\in U_0}^{\hat{r}_{ij}\in \text{top}K} \hat{r}_{ij} - \frac{1}{|U_1|} \sum_{i\in U_1}^{\hat{r}_{ij}\in \text{top}K} \hat{r}_{ij}|}{N}. \tag{14}$$

Some may argue that as different groups of users have different rating preferences, it is better to use group-based metrics that measure recommendation quality between different user groups. If we use r^{test} to denote a (user, item, rating) triplet that belongs to testing set, the absolute unfairness [23] can be expressed as follows: $1/N \sum_{j=1}^{N} ||E_{u\in U_0}[\hat{r}_{uv}^{\text{test}}] - E_{u\in U_0}[r_{uv}^{\text{test}}]| - |E_{u\in U_1}[\hat{r}_{uv}^{\text{test}}] - E_{u\in U_1}[r_{uv}^{\text{test}}]||$. However, this technique does not make sense in our specific case. Assume that absolute unfairness is also modified onto the top-K ranked items. In this case, we can only use the intersection of the top-K ranked items and the testing data to calculate recommendation performance, formulated as $1/N \sum_{j=1}^{N} ||E_{u\in U_0}[\hat{r}_{uv}^{\text{test}\cap \text{top}K}] - E_{u\in U_0}[r_{uv}^{\text{test}\cap \text{top}K}]| - |E_{u\in U_1}[\hat{r}_{uv}^{\text{test}\cap \text{top}K}] - E_{u\in U_1}[r_{uv}^{\text{test}\cap \text{top}K}]||$. Note that the number of ratings that satisfy both top$K$ and test is very sparse, which leads to inaccurate estimation. Therefore, we did not take equal performance as a metric in our case.

Practically, we selected Fairness@50 and Fairness@all to evaluate recommendation fairness, where a smaller Fairness value indicates better performance in terms of fairness in the recommendation results. We calculated Fairness@all and Fairness@50 for all user and item pairs that do not appear in the training data

Table 3 Performance on the MovieLens-100K dataset

Baselines	Base PMF model				Base LR-GCCF model			
	RMSE	AUC	Fairness@50	Fairness@all	RMSE	AUC	Fairness@50	Fairness@all
PMF/LR-GCCF	**0.9333**	0.6667	0.148	0.0750	**0.9277**	0.7414	0.1712	0.0792
NIPS_Non	0.9423	0.8389	0.1825	0.0572	–	–	–	–
NIPS_Pop	0.9401	0.9564	0.3815	0.1625	–	–	–	–
ICML_2019	1.044	0.5880	0.138	0.0575	0.0.9969	0.6669	0.1156	0.0516
FairCF_NIC	1.019	0.5723	0.1327	0.0706	0.9875	0.6617	0.1003	0.0433
FairCF	1.061	**0.5707**	**0.1213**	**0.0404**	0.9956	**0.6358**	**0.0713**	**0.0424**

Table 4 Performance on the MovieLens-1M dataset

Baselines	Base PMF model				Base LR-GCCF model			
	RMSE	AUC	Fairness@50	Fairness@all	RMSE	AUC	Fairness@50	Fairness@all
PMF/LR-GCCF	**0.8657**	0.7457	0.1612	0.0678	**0.8554**	0.7956	0.1566	0.0802
NIPS_Non	0.8696	0.7481	0.6427	0.0317	–	–	–	–
NIPS_Pop	0.8693	0.8994	0.9034	0.1524	–	–	–	–
ICML_2019	0.9219	0.5891	0.1562	0.0202	0.9150	0.5786	0.1392	0.0206
FairCF_NIC	0.8998	0.5226	0.1439	0.0226	0.8992	0.6064	0.1297	0.0357
FairCF	0.9279	**0.5221**	**0.1321**	**0.0158**	0.9012	**0.5719**	**0.1169**	0.0084

Table 5 Performance on the Lastfm-360K dataset

Baselines	Base PMF model				Base LR-GCCF model			
	RMSE	AUC	Fairness@50	Fairness@all	RMSE	AUC	Fairness@50	Fairness@all
PMF/LR-GCCF	**0.6712**	0.6271	0.1158	0.1447	**0.6698**	0.6280	0.0431	0.1629
NIPS_Non	0.6966	0.6725	0.1489	**0.0594**	–	–	–	–
NIPS_Pop	0.6865	0.6506	0.1772	0.1228	–	–	–	–
ICML_2019	0.6910	0.6198	0.1280	0.1726	0.6915	0.5567	0.0243	0.1396
FairCF_NIC	0.6840	0.6069	0.0898	0.1662	0.6879	0.5880	0.0306	0.1362
FairCF	0.7094	**0.5587**	**0.0469**	0.1661	0.7072	**0.5284**	**0.0112**	**0.1301**

for the MovieLens-1M and MovieLens-100K datasets. However, for the Lastfm-360K dataset, nearly 105 billion predicted unobserved interactions must be calculated. Thus, in consideration of time and memory costs, we only evaluated Fairness@all and Fairness@50 with 10000 selected users and 2500 items in the Lastfm-360K dataset.

4.3 Overall performance

The proposed FairCF framework is a flexible framework with any base recommendation model; we considered two base recommendation models, i.e., PMF and LR-GCCF. Tables 3–5 show the results obtained on the three datasets. Several observations can be made from the results shown in these three tables. First, the basic models demonstrated the best performance in terms of recommendation accuracy (i.e., RMSE); however, compared to the models with the fairness constraints, the basic models showed the worst performance in terms of the fairness metrics because realizing fairness involves removing sensitive information from the CF models and reducing recommendation accuracy.

Second, we found that NIPS_Non and NIPS_Pop showed the worse performance in terms of AUC compared to the other fairness models because they do not remove sensitive information from the embeddings, i.e., they directly eliminate the differences between all women's and men's interactions in the training data. Therefore, they only showed good performance in terms of Fairness@all among the considered fairness metrics. Note that NIPS_Non achieved better fairness results than NIPS_Pop because nonparity regularization (which is the key concept in NIPS_Non) is closer to Fairness@K and Fairness@all than population parity (which is the key concept in NIPS_Pop) [23].

Third, among the models with fairness constraints, the proposed FairCF demonstrated the best performance in terms of the fairness metrics. Compared to NIPS_Non and NIPS_Pop, FairCF showed better performance in terms of fairness of the user embeddings (AUC) and the fairness of the recommendation results (Fairness@50). On the Lastfm-360K dataset, FairCF based on PMF failed to perform well in terms of Fairness@all because most users tend to give higher or lower values in the Lastfm-360K dataset.

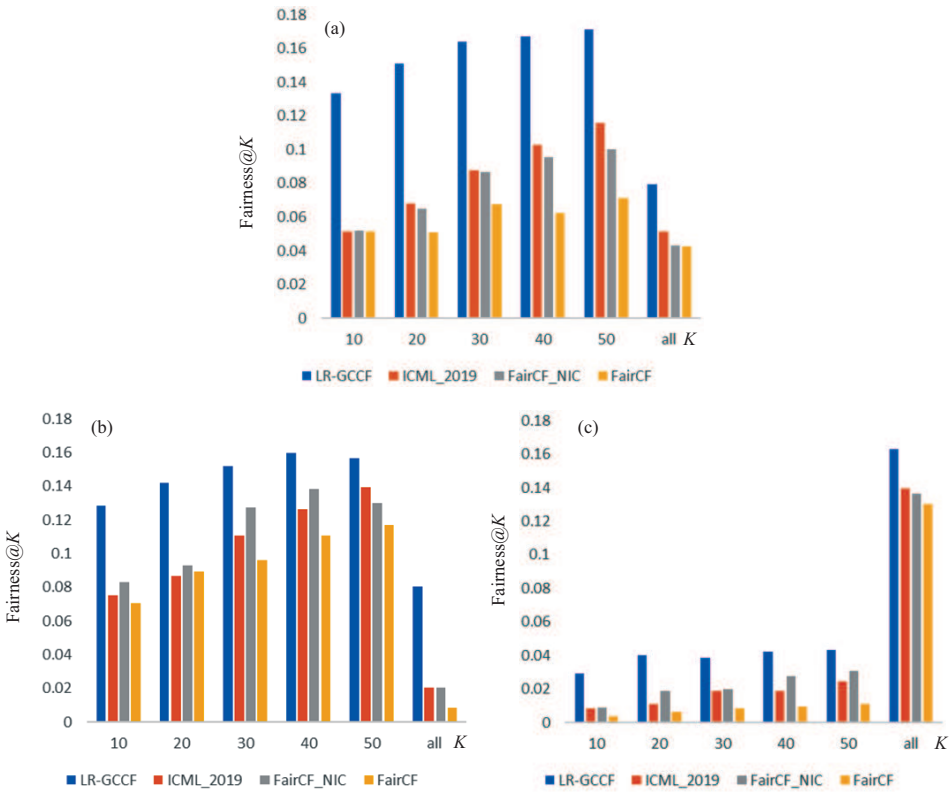

Figure 2 (Color online) Performance of Fairness@K with different K values. (a) Performance on MovieLens-100K dataset; (b) performance on MovieLens-1M dataset; (c) performance on Lastfm-360K dataset.

Thus, PMF tended to better optimize b_u and b_v to realize better recommendation accuracy. However, we only have fairness constraints on embeddings: thus, for low ratings on noninteractive items, unfairness appears to be influenced by b_u and b_v. Table 5 shows that Fairness@50 (i.e., calculating only the high predicted ratings for each user) still exhibits the better performance of both FairCF and FairCF_NIC on the Lastfm-360K dataset.

Fourth, we compared the proposed FairCF and FairCF_NIC. As shown in Tables 3–5, the proposed FairCF achieved better results in terms of the fairness metrics with a decrease of less than 5% RMSE compared to FairCF_NIC. These results demonstrate that the item classifier $\mathcal{D}_2$ with the pseudo item labels is effective relative to achieving fairness with CF-based models. There is a balance between fairness and recommendation accuracy. Thus, if a given amount of unfairness can be tolerated to improve accuracy, then FairCF_NIC can also be adopted.

Finally, we compared FairCF based on PMF and FairCF based on LR-GCCF, and we found that the LR-GCCF-based FairCF obtained better performance in terms of RMSE, Fairness@50, and Fairness@all. The reason for this is that the graph-based LR-GCCF model can model high-order correlations in a graph structure to alleviate the sparsity issue, and this model discovers more hidden features related to sensitive features. In addition, LR-GCCF has a better ability to generate embeddings that satisfy fairness requirements in adversarial learning; thus, the training process is more stable.

4.4 Detailed model analyses

LR-GCCF demonstrated better recommendation accuracy than PMF; thus, we selected LR-GCCF as the base model in the following detailed model analyses.

Performance of Fairness@K with different K values. We were interested in observing Fairness@K changes as K changes and whether FairCF can retain better performance in terms of Fairness@K. Here, we set K of Fairness@K in the range $[10, 20, 30, 40, 50]$ and calculated Fairness@all on four collaborative filtering models (i.e., LR-GCCF, ICML_2019, FairCF_NIC, and FairCF) and three datasets. As shown in Figure 2, the proposed FairCF demonstrated the best fairness performance in all cases, which indicates that FairCF is both stable and effective.

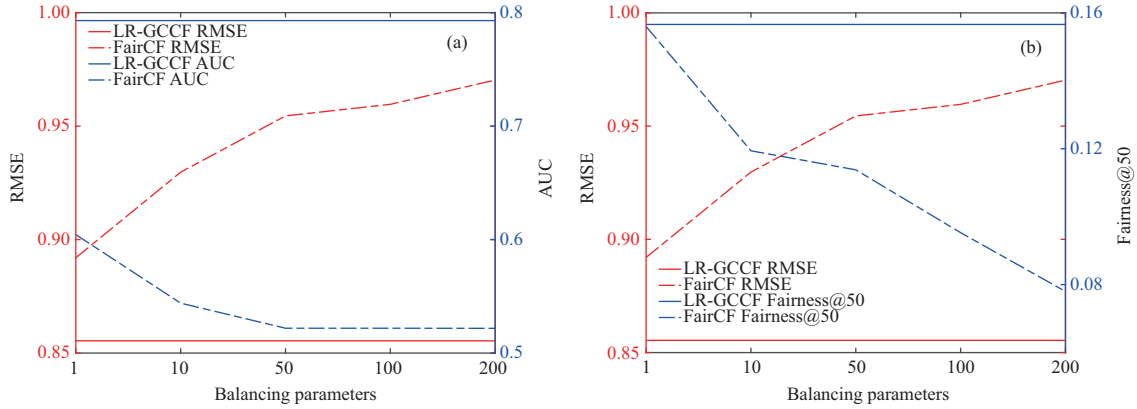

Figure 3 (Color online) Accuracy and fairness results obtained under different settings for parameters β and γ on the MovieLens-1M dataset. (a) Accuracy and AUC; (b) accuracy and Fairness@50.

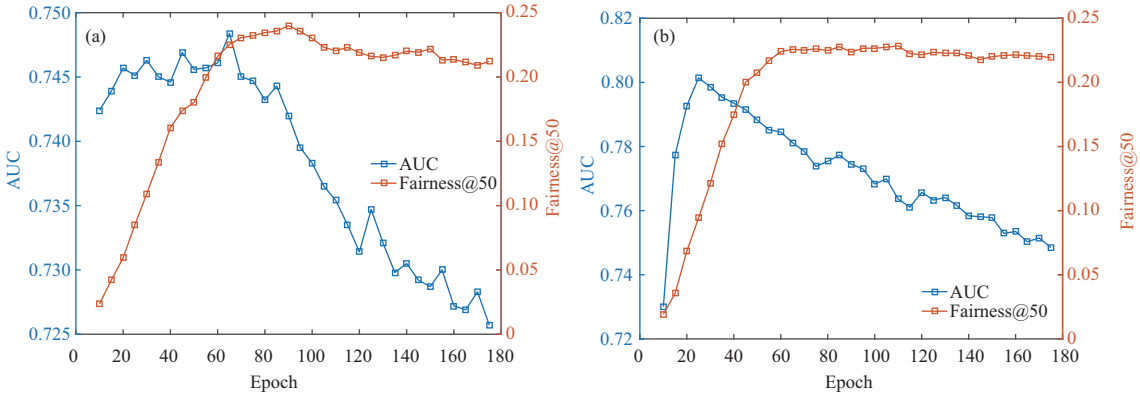

Figure 4 (Color online) Different fairness metrics with different epochs on the MovieLens-1M dataset. (a) Training process of PMF; (b) training process of LR-GCCF.

Trade-off between recommendation accuracy and fairness. Balancing hyperparameters β and γ allowed us to control the trade-off between recommendation accuracy and fairness. As mentioned previously, parameters β and γ were set to 10 (Subsection 4.3) on the MovieLens-1M dataset. We recorded the results obtained on the MovieLens-1M dataset with different values for balancing parameters β and γ in Figure 3. To facilitate a better comparison, we also set the same values for β and γ. Specifically, we conducted experiments and compared the results when the β and γ values were varied in the range $[1, 10, 50, 100, 200]$. Here, we used the RMSE metric to indicate accuracy and (Fairness@50, AUC) to represent fairness. In Figure 3, the blue and red horizontal lines represent accuracy and fairness, respectively, for LR-GCCF. Note that these lines are not related to β and γ. In terms of FairCF, Figure 3 shows that larger values for the balancing parameters β and γ resulted in better performance in terms of fairness and worse performance in terms of accuracy.

Connections between AUC metric and Fairness@K metrics. Two types of fairness metrics were considered in our experiments, i.e., AUC based on the user embedding and Fairness@K based on the recommendation results. We wondered whether a relationship exists between these two metrics. Here, the intuitive idea is that if AUC tends to be 0.5, the embeddings of two genders will be in the same data distribution; thus, Fairness@K tends to be 0. In Figure 4, we recorded these fairness metrics for the MovieLens-1M obtained every five epochs with the baseline PMF and LR-GCCF models. Several observations can be made from Figure 4. First, AUC and Fairness@50 increased because the recommender systems utilized gender information to improve accuracy when training begins. Second, as the recommendation models began to overfit the training data, Fairness@50 and AUC decreased with the PMF and LR-GCCF models.

Finally, we observed that these two metrics will not decrease as small as a random situation, e.g., AUC decreases to 0.5 or Fairness@50 decreases to 0, because gender information still has an impact on recommendation models. We found that AUC and Fairness@50 demonstrate a similar tendency, i.e., the

Table 6 Performance with multiple sensitive attributes

Baselines	Attribute	Base PMF model			Base LR-GCCF model		
		RMSE	AUC	Micro-F1	RMSE	AUC	Micro-F1
PMF/LR-GCCF	None	**0.8657**	0.7457	0.3956	**0.8554**	0.7956	0.4031
ICML_2019	Gender	0.9219	0.5891	0.3518	0.9150	0.5786	0.3591
ICML_2019	Age	0.9346	0.5901	0.3493	0.9230	0.6781	0.3501
ICML_2019	Multiple	0.9371	0.5571	0.3485	0.9257	0.5640	0.3493
FairCF	Gender	0.9279	0.5221	0.3543	0.9012	0.5719	0.3526
FairCF	Age	0.9246	0.6013	0.3501	0.9035	0.6351	0.3493
FairCF	Multiple	0.9301	**0.5163**	**0.3469**	0.9126	**0.5572**	**0.3485**

degree of response to unfairness varies.

Extension to multiple sensitive attributes. As mentioned in Section 1, our primary contribution lies in discovering the correlations of users and items for fairness modeling in recommendation systems and proposing a simple solution to address this issue. The experiments conducted on a single sensitive attribute (i.e., gender) demonstrated the superiority of the proposed technique. We also extend the proposed FairCF to multiple sensitive attributes to prove its effectiveness. Here, by utilizing a composition of filters $\mathcal{F} = [\mathcal{F}^c]_{c=1}^C$, we transformed the embeddings based on C sensitive attributes [24,39]. The filtered embeddings are represented as $\boldsymbol{f}_i = \frac{\sum_{c=1}^C \mathcal{F}^c(\boldsymbol{e}_i)}{C}$, where $\mathcal{D}_1 = [\mathcal{D}_1^c]_{c=1}^C$ and $\mathcal{D}_2 = [\mathcal{D}_2^c]_{c=1}^C$ represent user adversaries and item adversaries, respectively. In addition, $\boldsymbol{S}^c$ and $\boldsymbol{P}^c$ represent user sensitive labels and pseudo item labels for sensitive attribute c, respectively. As a result, the minimax game can be formulated as follows:

$$\min_{\mathcal{E},\mathcal{F}} \max_{\mathcal{D}_1,\mathcal{D}_2} L_{\mathcal{E}}(\boldsymbol{F}_U, \boldsymbol{F}_V) - \sum_{c=1}^{C} [\beta^c L_{\mathcal{D}_1^c}(\boldsymbol{F}_U, \boldsymbol{S}^c) + \gamma^c L_{\mathcal{D}_2^c}(\boldsymbol{F}_V, \boldsymbol{P}^c)], \tag{15}$$

where β^c and γ^c are balancing parameters for sensitive attribute c.

Following (15), we conducted further experiments on the MovieLens-1M dataset to evaluate performance with multiple sensitive attributes. Here, we treated the gender and age attributes as sensitive attributes. In other words, the fairness-aware models under multiple sensitive attributes simultaneously utilized a gender filter and age filter to eliminate gender and age information in an adversarial manner. Following [24, 39], we randomly selected 80% of the users, trained a classification model by taking the learned representations of the 80% of users, and calculated the classification performance on the 20% test users to measure the fairness performance.

In addition, we used AUC for the binary sensitive attribute gender. For the sensitive attribute age, following previous studies [24, 39], we used micro-averaged F1, which is often used for evaluating multi-class classification. Note that smaller AUC or micro-averaged F1 values indicate less information leakage for the corresponding sensitive attribute and better performance on fairness, respectively. The hyperparameters for gender are listed in Table 2. For the hyperparameters for age, we set balancing parameters $\beta^{\text{age}} = 1$ and $\gamma^{\text{age}} = 20$. In addition, we set times of $(\mathcal{E}^{\text{age}}, \mathcal{F}^{\text{age}})$, $\mathcal{D}_1^{\text{age}}$, and $\mathcal{D}_2^{\text{age}}$ during one epoch to one, 40, and 40, respectively.

The results are shown in Table 6. As can be seen, the proposed FairCF was able to handle multiple sensitive attributes and outperformed ICML_2019 in terms of both fairness and accuracy. Two important observations can be taken from the results shown in Table 6. First, the models that focus on gender (age) fairness also performed better on age (gender) fairness than the models without fairness goals (i.e., PMF and LR-GCCF). Second, the fairness-aware models under the multiple sensitive setting demonstrated better fairness results compared to the base models. These findings indicate that the different sensitive attributes (i.e., age and gender) are correlated, and the multiple sensitive attributes setting can exploit the correlations between different sensitive attributes to eliminate more sensitive information.

5 Conclusion and future work

In this paper, we have proposed the FairCF framework for fairness consideration in CF-based recommendation systems. We have argued that as user and item embeddings in CF models are collaboratively correlated, only eliminating unfairness at the user embeddings is not satisfactory. Thus, we designed fairness constraints in the fair embedding space, and user and item classifiers were utilized in the proposed

framework to fit the fairness constraints. Experimental results obtained on three real-world datasets clearly demonstrate that our proposed FairCF achieved over 5% improvements on fairness with less than 4% decrease on recommendation accuracy, compared to other fairness-aware CF models (NIPS_Non, NIPS_Pop, ICML_2019).

In the future, an important problem must be addressed: most existing studies related to fairness have ignored modeling the correlations between different sensitive attributes. Such correlations can be valuable in many situations related to fairness, e.g., existing studies rarely considered the practical scenario of missing values for sensitive attributes. We consider that it would be promising to infer missing values with correlations to address this problem. Thus, extending the proposed framework by mining attribute correlations is a natural direction for future work.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61972125, U19A2079, 61725203, 61732008, 62006066) and Fundamental Research Funds for the Central Universities (Grant No. JZ2020HGPA-0114). Le WU greatly thanks the support of Young Elite Scientists Sponsorship Program by CAST and ISZS.

References

1 van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. In: Proceedings of Neural Information Processing Systems, 2013. 2643–2651
2 Wu J W, Shen L W, Guo W N, et al. Code recommendation for android development: how does it work and what can be improved? Sci China Inf Sci, 2017, 60: 092111
3 Ying R, He R N, Chen K F, et al. Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 974–983
4 Chen H H, Jin H, Cui X L. Hybrid followee recommendation in microblogging systems. Sci China Inf Sci, 2017, 60: 012102
5 Mnih A, Salakhutdinov R R. Probabilistic matrix factorization. In: Proceedings of Neural Information Processing Systems, 2008. 1257–1264
6 Rendle S, Freudenthaler C, Gantner Z, et al. BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009. 452–461
7 He X N, Liao L Z, Zhang H W, et al. Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, 2017. 173–182
8 Lei C, Le W, Richang H, et al. Revisiting graph based collaborative filtering: a linear residual graph convolutional network approach. In: Proceedings of AAAI Conference on Artificial Intelligence, 2020. 27–34
9 Wu L, Yang Y H, Zhang K, et al. Joint item recommendation and attribute inference: an adaptive graph convolutional network approach. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020. 679–688
10 Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer, 2009, 42: 30–37
11 Lahoti P, Gummadi K P, Weikum G. iFair: learning individually fair data representations for algorithmic decision making. In: Proceedings of the 35th International Conference on Data Engineering, 2019. 1334–1345
12 Wu C H, Wu F Z, Wang X T, et al. Fairness-aware news recommendation with decomposed adversarial learning. In: Proceedings of AAAI Conference on Artificial Intelligence, 2021. 4462–4469
13 Sweeney L. Discrimination in online ad delivery. SSRN J, 2013, 11: 10–29
14 Pedreshi D, Ruggieri S, Turini F. Discrimination-aware data mining. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008. 560–568
15 Kamiran F, Calders T. Classifying without discriminating. In: Proceedings of the 2nd International Conference on Computer, Control and Communication, 2009. 1–6
16 Luong B T, Ruggieri S, Turini F. k-NN as an implementation of situation testing for discrimination discovery and prevention. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011. 502–510
17 Zhang B H, Lemoine B, Mitchell M. Mitigating unwanted biases with adversarial learning. In: Proceedings of AAAI/ACM Conference on AI, Ethics, and Society, 2018. 335–340
18 Kamishima T, Akaho S, Sakuma J. Fairness-aware learning through regularization approach. In: Proceedings of the 11th International Conference on Data Mining Workshops, 2011. 643–650
19 Zemel R, Wu Y, Swersky K, et al. Learning fair representations. In: Proceedings of the 30th International Conference on International Conference on Machine Learning, 2013. 325–333
20 Edwards H, Storkey A J. Censoring representations with an adversary. In: Proceedings of International Conference on Learning Representations, 2016
21 Beutel A, Chen J L, Zhao Z, et al. Data decisions and theoretical implications when adversarially learning fair representations. In: Proceedings of Workshop on Fairness, Accountability, and Transparency in Machine Learning, 2017
22 Madras D, Creager E, Pitassi T, et al. Learning adversarially fair and transferable representations. In: Proceedings of the 35th International Conference on Machine Learning, 2018. 3381–3390
23 Yao S R, Huang B. Beyond parity: fairness objectives for collaborative filtering. In: Proceedings of the 31st Conference on Neural Information Processing System, 2017. 2921–2930
24 Bose A J, Hamilton W. Compositional fairness constraints for graph embeddings. In: Proceedings of the 36th International Conference on Machine Learning, 2019. 715–724
25 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of Neural Information Processing Systems, 2014. 2672–2680
26 Covington P, Adams J, Sargin E. Deep neural networks for youtube recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, 2016. 191–198
27 Paparrizos I, Cambazoglu B B, Gionis A. Machine learned job recommendation. In: Proceedings of the 5th ACM Conference on Recommender systems, 2011. 325–328
28 Wu L, Ge Y, Liu Q, et al. Modeling the evolution of users' preferences and social links in social networking services. IEEE Trans Knowl Data Eng, 2017, 29: 1240–1253

29 Wu L, Chen L, Hong R C, et al. A hierarchical attention model for social contextual image recommendation. IEEE Trans Knowl Data Eng, 2020, 32: 1854–1867
30 Mehrabi N, Morstatter F, Saxena N, et al. A survey on bias and fairness in machine learning. ACM Comput Sur, 2021, 54: 1–35
31 Mukherjee D, Yurochkin M, Banerjee M, et al. Two simple ways to learn individual fairness metrics from data. In: Proceedings of the 37th International Conference on Machine Learning, 2020. 7097–7107
32 Kusner M J, Loftus J, Russell C, et al. Counterfactual fairness. In: Proceedings of Neural Information Processing Systems, 2017. 4066–4076
33 Hardt M, Price E, Srebro N. Equality of opportunity in supervised learning. In: Proceedings of Neural Information Processing Systems, 2016. 3315–3323
34 Binns R. Fairness in machine learning: lessons from political philosophy. In: Proceedings of Conference on Fairness, Accountability, and Transparency, 2018. 149–159
35 Dai E Y, Wang S H. Say no to the discrimination: learning fair graph neural networks with limited sensitive attribute information. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021. 680–688
36 Zhu Z W, Hu X, Caverlee J. Fairness-aware tensor-based recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018. 1153–1162
37 Beutel A, Chen J L, Doshi T, et al. Fairness in recommendation ranking through pairwise comparisons. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019. 2212–2220
38 Islam R, Keya K N, Zeng Z Q, et al. Debiasing career recommendations with neural fair collaborative filtering. In: Proceedings of Web Conference, 2021. 3779–3790
39 Le W, Lei C, Shao P Y, et al. Learning fair representations for recommendation: a graph-based perspective. In: Proceedings of Web Conference, 2021. 2198–208
40 Ekstrand M D, Tian M, Azpiazu I M, et al. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In: Proceedings of Conference on Fairness, Accountability and Transparency, 2018. 172–186
41 Lambrecht A, Tucker C. Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads. Manage Sci, 2019, 65: 2966–2981
42 Wang Y X, Wu C S, Herranz L, et al. Transferring GANs: generating images from limited data. In: Proceedings of European Conference on Computer Vision, 2018. 220–236
43 Harper F M, Konstan J A. The movielens datasets: history and context. ACM Trans Interact Intell Syst, 2015, 5: 1–19
44 Herrada Ò C. Music recommendation and discovery in the long tail. Dissertation for Ph.D. Degree. Barcelona: Universitat Pompeu Fabra, 2009