Counterfactual Explanations for Recommendation Bias

Leonidas Zafeiriou[1], Panayiotis Tsaparas[1,2], and Evaggelia Pitoura[1,2]

[1] Deptment of Computer Science & Engineering, University of Ioannina, Greece
[2] Archimedes/Athena Recerch Center, Greece

Abstract. Today, we rely heavily on automated recommendation algorithms for assisting us in making several decisions. These algorithms are trained on large quantities of user interaction data, and as a result they incorporate various biases of the data in their recommendations. It is important to understand the origins of recommendation biases, however, this is becoming increasingly difficult, given the complexity of the recommenders. To address model complexity, researchers try to provide explanations for the behavior of the algorithms, such as counterfactual explanations. In this work, we consider explanations for recommendation bias, and we generalize counterfactual explanations to handle groups of users and items. We then consider a random-walk based recommender, and we propose efficient algorithms for computing the counterfactual explanations. We perform an experimental evaluation of our algorithms using both real and synthetic data.

Keywords: bias · fairness · explanations.

1 Introduction

Today, we rely heavily on automated recommendation algorithms for assisting us in making several decisions, such as the content we consume, the items we buy, or the careers we pursue. These algorithms use sophisticated machine learning techniques that are trained on large quantities of user interaction data. As a result, they incorporate various *biases* in their recommendations, where certain groups of users or items are treated differently. Although these biases are to some extent integral to the algorithms in order to make personalized recommendations, they can also lead to unfair treatment of sensitive groups.

Fairness and bias in recommendations is a problem that has received significant attention in the past years [10, 22]. Different definitions of fairness have been adopted, depending on whether fairness is defined with respect to consumers or producers [3]. In most definitions, we assume that there are groups of users and/or items, defined based on sensitive attributes such as gender, religion or age for users, and type of content for items. The recommender should treat the groups fairly, e.g., producing recommendations of equal quality for the two user groups, or representing proportionally the items in different categories. When this is not achieved, we consider the recommender to be biased.

Understanding recommendation biases is important in monitoring the health of the recommendation system and ensuring fairness. However, given the complexity of recommendation algorithms, this is becoming increasingly difficult. To address the complexity of "black box" systems, there is a strong research movement towards producing different types of explanations for the behavior of algorithms, including recommendation algorithms [24]. One type of explanations is counterfactual explanations [21], where we look for a small number of changes in the input data that will achieve a desired change in the output of the algorithm, e.g., change the classification of a data point.

In this work, we consider counterfactual explanations for recommendation bias. Previous work on counterfactual explanations for recommendations focused on explaining the decisions of the recommender for specific user-item pairs [17, 7, 18]. Given that bias is defined with respect to groups of users and items instead of specific user-item pairs, we need to generalize the definition of counterfactual explanations to handle this case. We consider different types of explanations. First, we consider individual users, and we seek explanations as to why a user does not get enough recommendations from a specific item category. We extend these explanations to the case where we have a group of users instead of an individual user. We then consider individual items, and we look for explanations as to why they do not get recommended to a specific group of users. Again, we extend these explanations to the case where we have an item category rather than a single item.

We consider a graph-based random walk recommender, and we propose algorithms for computing the counterfactual explanations. Our algorithms exploit Linear Algebra tools to efficiently estimate the effect of a change to the recommendations, and they can be applied to large datasets. We perform an experimental evaluation of our algorithms using a real Movies dataset, as well as synthetic data. Our experiments study the hardness of producing explanations for different cases, and provide understanding of the dataset characteristics that affect the explanations.

In summary, in this work we make the following contributions:

- We define and formalize the novel problem of counterfactual explanations for different types of recommendation bias.
- We propose efficient algorithms for computing counterfactual explanations for a random walk recommender that scale for large datasets.
- We evaluate quantitatively and qualitatively our algorithms on real and synthetic datasets.

The rest of the paper is structured as follows. In Section 2, we provide the definitions for our problems. In Section 3, we present efficient algorithms for producing counterfactual explanations for recommendation bias. In Section 4, we present our experimental evaluation. Section 5 presents the related work, and Section 6 concludes the paper.

2 Definitions

We are given as input a set of users $\mathcal{U}$ and a set of items $\mathcal{I}$ and a user-item matrix D with the preferences of the users over the items. We also have a recommender R_D that is trained on the matrix D, which, given a pair $(u, i) \in \mathcal{U} \times \mathcal{I}$, it outputs a score $R_D(u, i)$ that is the estimation of the preference of user u for the item i.

We assume that we can define groups of users and items based on the attributes of the users and items respectively. For example, we may partition users into groups based on gender, age, or residence. Similarly, if the items are movies, we may define groups of movies based on genre, or on release date. We are interested in defining biases that the recommender may have towards specific groups of items or users, and provide explanations for them.

We first consider the bias of the recommender in the estimated ratings for an individual user for a specific group of items. For example in the user-movie scenario, we want to explain why the estimated ratings for a specific user are on average lower for Romance movies, than for Action. Formally, let u be a specific user, and let $I \subset \mathcal{I}$ denote the target item group. Let $\overline{I} = \mathcal{I} \setminus I$ denote the items not in the group. We define $R_D(u, I) = \frac{1}{|I|} \sum_{i \in I} R_D(u, i)$ to be the average estimated score of the recommender for user u for the items in I, and $R_D(u, \overline{I})$ for the complement group. We define the preference ratio of recommender R_D for item group I for user u as:

$$B_{R_D}(I|u) = \frac{R_D(u, I)}{R_D(u, \overline{I})}$$

Given a target value θ (usually $\theta = 1$), we say that the recommender is *biased against* group I in the recommendations to user u, if $B_{R_D}(I|u) < \theta$. The value θ is input to our problem, and it determines our sensitivity to the input bias. We will usually set $\theta = 1$; the recommender is biased if the estimated ratings it produces for user u for group I are on average lower than those for the complement group.

We seek *counterfactual* explanations for the bias of the recommender: changes in the ratings D_u of user u that will result in an increase of the preference ratio for group I. Formally, an explanation is a subset $E_u \subset D_u$ of the ratings of u, such that, if removed from D, the resulting recommender $R_{D|E_u}$ is not biased. The size of the set E_u is the *complexity* of the explanation. The goal is to find small explanations that explain the bias.

We thus have the following problem definition:

Problem 1 (Individual User Bias Explanation). Given preference matrix D, a recommender R_D, a user u, a target group of items I, and a target bias value θ, find the minimum explanation E_u such that, $B_{R_{D|E_u}}(I|u) \geq \theta$.

We can extend the definition of bias to the case where we have a group of users instead of a single user. In the user-movie example, we want to explain why the average ratings of Male users for Romance movies are lower than for Action movies. Let $U \subset \mathcal{U}$ denote the target user group. For a target group of

items I, we define $R_D(U, I) = \frac{1}{|U|} \sum_{u \in U} R_D(u, I)$ to be the average estimated score for the item group I, for user group U. We define the preference ratio of recommender R_D for item group I, for user group U as:

$$B_{R_D}(I|U) = \frac{R_D(U, I)}{R_D(U, \overline{I})}$$

Given a target value θ, we say that the recommender is biased against item group I in the recommendations to U, if $B_{R_D}(I|U) < \theta$. We seek again counterfactual explanations for the bias of the recommender. Let D_U denote the set of ratings from users in U. The explanation is a subset $E_U \subset D_U$. We have the following problem definition:

Problem 2 (User Group Bias Explanation). Given preference matrix D, a recommender R_D, a target group of users U, a target group of items I, and a target bias value θ, find the minimum explanation E_U, such that $B_{R_{D|E_U}}(I|U) \geq \theta$.

We now turn our attention to the item side, and we consider the bias of the recommender in the estimated ratings that an individual item i receives from a group of users U. In our user-movie example, we look at a specific movie, and we want to explain, why this movie receives on average lower ratings from Male users than from Female users.

Formally, let i be a specific item, let $U \subset \mathcal{U}$ denote the target user group, and let $\overline{U} = \mathcal{U} \setminus U$ denote the users not in the group. We define $R_D(U, i) = \frac{1}{|U|} \sum_{u \in U} R_D(u, i)$ to be the average estimated score of the recommender for group U for the item i. We define the preference ratio of recommender R_D f for item i, or user group U as:

$$B_{R_D}(U|i) = \frac{R_D(U, i)}{R_D(\overline{U}, i)}$$

Given a target value θ, we say that the recommender is biased against the group U in the recommendations for i, if $B_{R_D}(U|i) < \theta$. When $\theta = 1$, the recommender is biased against user group U if the ratings it estimates for the users in U for item i are on average lower than those for the remaining users.

We seek again counterfactual explanations for the bias of the recommender. Let D_i denote the users that have rated item i, and $U_i = D_i \cap U$ denote the users in U that have rated item i. An explanation is a subset $E_i \subseteq U_i$ of the ratings from users in U_i. We thus have the following definition:

Problem 3 (Individual Item Bias Explanation). Given preference matrix D, a recommender R_D, an item $i \subset \mathcal{I}$, a target group of users U, and a target bias value θ, find the minimum explanation E_i such that $B_{R_{D|E_i}}(U|i) \geq \theta$.

We will also consider the the bias of the recommender in the estimated ratings that a group of items I receives from a group of users U. In our example, we want to explain why the Romance category receives lower scores for Males than

it receives for Females. We define the preference ratio of recommender R_D for item group I, for group U as:

$$B_{R_D}(U|I) = \frac{R_D(U,I)}{R_D(\overline{U},I)}$$

For a target value θ, we say that R_D is biased against user group U in the recommendations for item group I, when $B_{R_D}(U|I) < \theta$. As in the case of the item bias explanations we seek an explanation $E_U \subset D_U$, as a subset of the ratings of the group U, that, if removed, will result in an increase in $B_{R_D}(U|I)$.

Problem 4 (Item Group Bias Explanation). Given user-item preference matrix D, a recommender R_D, a target item group I, a target user group U, and a target value θ, find the minimum explanation E_U such that $B_{R_{D|E_U}}(U|I) \geq \theta$.

Our definitions of bias and explanations for bias are general, and they can be applied to any recommender. However, clearly, the exact explanations depend on the recommendation algorithm R_D. In the next section, we present a random-walk recommendation algorithm R_D that we will consider in this work, and efficient algorithms for computing individual and group explanations.

3 Explanations in Graph Recommenders

3.1 The recommendation algorithm

As our recommendation algorithm, we will use the RecWalk algorithm [9]. We view the user-item matrix D as a bipartite graph G with adjacency matrix: $A_G = \begin{pmatrix} 0 & D \\ D^T & 0 \end{pmatrix}$ The RecWalk algorithm estimates the scores for user-item pairs by performing random walks on the graph G. Let $H = Diag(A_G\mathbb{1})^{-1}A_G$, where $\mathbb{1}$ is the vector of all ones, be the transition probability of a simple random walk on the user-item bipartite graph. Let M_I be an inter-item transition probability matrix that captures relations between items, and define matrix M as: $M = \begin{pmatrix} I & 0 \\ 0 & M_I \end{pmatrix}$. The overall transition probability matrix of RecWalk is defined as $P = \alpha\, H + (1-\alpha)\, M$ where α captures the relative contribution of each of the two components in the random walk.

To compute recommendations for a user u, we perform a *personalized random walk* rooted at u. At each step the random walk with probability $(1-\gamma)$ transitions according to matrix P, while with probability γ it restarts from node u. Given the stationary distribution $\mathbf{p}_u$ of the random walk, the estimated score of item i for user u is computed as $R_D(u,i) = \mathbf{p}_u(i)$. Note that we can use matrix P to perform a personalized random walk rooted at any node x in the graph, either user or item, and produce a probability $p_x(y)$ for any node in the graph, user or item. We will make use of these probabilities in our computations below.

Given this graph view of the data and the algorithm, the counterfactual explanations that we consider consist of (directed) edges emanating users, which, if deleted, will result in an increase in the target ratings.

3.2 Estimating the effect of edge removals

We begin with the computation of the change in the estimated rating $R_D(u,i)$ for a user-item pair (u,i), when deleting a user-item edge (x,y) from the graph. Note that x may be different from u, and y may be different form i. We want to estimate

$$\Delta\left(u,i,(x,y)\right) = R_{D|(x,y)}(u,i) - R_D(u,i)$$

We will provide analytical formulas for this computation which we will then use for the different problems we consider in this paper. For simplicity we assume that $\alpha = 0$, but our formulas can easily be extended to the case that $\alpha \neq 0$.

Recall that $R_D(u,i) = \mathbf{p}_u(i)$ denotes the personalized random walk probability of user u for item i. We use $R_{D|(x,y)}(u,i) = \mathbf{p}_u(i|(x,y))$ to denote the probability of user u for item i after the removal of edge (x,y). We want to estimate $\Delta(u,i,(x,y)) = \mathbf{p}_u(i|(x,y)) - \mathbf{p}_u(i)$. We can prove the following:

$$\Delta\left(u,i,(x,y)\right) = \mathbf{p}_u(x)\Lambda(x,i,(x,y)) \tag{1}$$

where

$$\Lambda\left(x,i,(x,y)\right) = \frac{\frac{1-\gamma}{\gamma}\left(\frac{1}{|D_x|}\sum_{j\in D_x}\mathbf{p}_j(i) - \mathbf{p}_y(i)\right)}{|D_x| - 1 - \frac{1-\gamma}{\gamma}\left(\frac{1}{|D_x|}\sum_{j\in D_x}\mathbf{p}_j(x) - \mathbf{p}_y(x)\right)} \tag{2}$$

In Equation 2, D_x denotes the outgoing edges from node x, and $\mathbf{p}_i$ the stationary distribution of the personalized random walk rooted at item i

This formula can be extended to the case where we remove multiple edges from the node x. Let $E_x \subset \{(x,y) : y \in D_x\}$ denote the set of edges removed from x (we assume that at least one edge from x remains in the graph). Abusing the notation, we will also use E_x to denote the set of neighbors of x from which we remove the edges. We can estimate $\Delta(u,i,E_x) = \mathbf{p}_u(i|E_x) - \mathbf{p}_u(i)$ as follows:

$$\Delta(u,i,E_x) = \mathbf{p}_u(x)\Lambda(x,i,E_x) \tag{3}$$

where

$$\Lambda(x,i,E_x) = \frac{\frac{1-\gamma}{\gamma}\left(\frac{1}{|D_x|}\sum_{j\in D_x}\mathbf{p}_j(i) - \frac{1}{|E_x|}\sum_{j\in E_x}\mathbf{p}_j(i)\right)}{\frac{|D_x|-|S_x|}{|S_x|} - \frac{1-\gamma}{\gamma}\left(\frac{1}{|D_x|}\sum_{j\in D_x}\mathbf{p}_j(x) - \frac{1}{|E_x|}\sum_{j\in E_x}\mathbf{p}_j(x)\right)} \tag{4}$$

Consider now the case where we want to compute explanations for the individual user bias for user u towards item group I. Abusing the notation, let $\mathbf{p}_u(I) = \sum_{i\in I}\mathbf{p}_u(i)$; therefore, $R(u,i) = \frac{1}{|I|}\mathbf{p}_u(I)$. Let $\Delta(u,I,E_u) = R_{D|E_u}(u,I) - R_D(u,I)$. Using Equations 3 and 4, we have:

$$\Delta(u,I,E_u) = \mathbf{p}_u(u)\Lambda(u,I,E_u) \tag{5}$$

$$\Lambda(u,I,E_u) = \frac{\frac{1-\gamma}{\gamma}\left(\frac{1}{|D_u|}\sum_{j\in D_u}\frac{1}{|I|}\mathbf{p}_j(I) - \frac{1}{|E_u|}\sum_{j\in E_u}\frac{1}{|I|}\mathbf{p}_j(I)\right)}{\frac{|D_u|-|E_u|}{|E_u|} - \frac{1-\gamma}{\gamma}\left(\frac{1}{|D_u|}\sum_{j\in D_u}\mathbf{p}_j(u) - \frac{1}{|E_u|}\sum_{j\in E_u}\mathbf{p}_j(u)\right)} \tag{6}$$

Consider now a group of users U and an item i. The effect of removing a set of edges E_x from a node x, $\Delta(U, i, E_x) = R_{D|E_x}(U, i) - R_D(U, i)$ can be estimated as

$$\Delta(U, i, E_x) = \mathbf{p}_U(x)\Lambda(x, i, E_x) \tag{7}$$

where $\mathbf{p}_U(x) = \frac{1}{U}\sum_{u \in U} \mathbf{p}_u(x)$ and $\Lambda(x, i, E_x)$ is computed as in Equation 4.

When considering a group of users U and a group of items I, the effect of removing a set of edges E_x from a node x, $\Delta(U, I, E_x) = R_{D|E_x}(U, I) - R_D(U, I)$ can be estimated as

$$\Delta(U, I, E_x) = \mathbf{p}_U(x)\Lambda(x, I, E_x) \tag{8}$$

where $\Lambda(x, I, E_x)$ is computed as in Equation 6.

The key observation is that the computation of the Δ-values relies on the computation of quantities such as $\mathbf{p}_j(u)$, $\mathbf{p}_j(i)$ and $\mathbf{p}_j(I)$. We can efficiently compute these terms in a single Pagerank-like computation, by adding absorbing nodes to the graph, and performing an absorbing random walk. We describe the details in the Supplementary Material available at our github repository online[3].

3.3 Algorithms for computing bias explanations

Individual user bias explanations. In the case of individual user explanations, we want to explain why for a user u the scores of the recommender for the target group I are lower than those for the complement group $\overline{I}$. To find these explanations we look for the edges $E_u \subset D_u$ emanating from u whose removal will maximize $\Delta(u, I, E_u)$. We use a *greedy* algorithm for this task. We incrementally build the set E_u, each time adding the edge (u, v) that maximizes the gain $\text{gain}(u, v) = \Delta(u, I, E_u \cup \{(u, v)\}) - \Delta(u, I, E_u)$. Note that we can implement the Greedy algorithm very efficiently. We compute for every node i in the graph the quantities $\mathbf{p}_i(u)$ and $\mathbf{p}_i(I)$ only once, at the beginning of the algorithm. Then at any iteration of the algorithm we can compute $\text{gain}(u, v)$ with simple mathematical operations using Equation 5. We will refer to this algorithm as GREEDY.

For comparison, we will also consider the algorithm that computes $\Delta(u, I, (u, v))$ for each edge (u, v), sorts the edges in decreasing order of the Δ-values, and selects them in that order. This algorithm is more efficient as it makes only one computation initially. We will refer to this algorithm as SORT.

Individual Item Explanations. In the case of individual item explanations, we want to explain why for a specific item, the recommendation algorithm estimates lower scores from group U than $\overline{U}$. The explanations consist of edges from the users in U_i, the users in the group U that have rated item i. The algorithm computes $\Delta(U, i, (x, y))$ for each edge (x, y), $x \in U_i$, $y \neq i$. It then sorts the edges according to these values, and returns the top edges that achieve the target bias value θ.

[3] https://github.com/lezaf/BiasExplain

User-group bias explanations. In the case of user-group explanations, we want to explain why for the user group U the scores of the recommender R for the item group I are lower than those for the complement item group $\overline{I}$. The explanation E_U consists of a set of changes in the ratings of the users in U that will correct the bias of the recommender. We will consider three algorithms for constructing the explanations, each leading to a qualitatively different type of explanations.

The first algorithm looks for the best set of edges from U to remove. Note that it is easy to show that we only need to consider edges (u, v) to the complement item group $\overline{I}$, that is, $v \in \overline{I}$; otherwise, we decrease $R_D(U, I)$. The algorithm computes the value $\Delta(U, I, (u, i))$ for each edge (u, i), where $u \in U$, and $i \in \overline{I}$, it sorts the edges, and selects the top ones that achieve the target bias value θ. We will refer to this algorithm as EdgeExplain.

The second algorithm builds the explanation by selecting *users* from U, and removing all their edges to the complement group $\overline{I}$. The explanation in this case is a set of users rather than a set of edges. To compute the explanation, for each user $u \in U$, let $E_u(\overline{I})$ denote the set of edges from u to the group $\overline{I}$. The algorithm estimates $\Delta(u, I, E_u(\overline{I}))$ for all users $u \in U$, sorts them according to the Δ-values, and returns the top ones that achieve the target bias goal. We will refer to this algorithm as UserExplain.

The third algorithm builds the explanation by selecting *items* from the complement group $\overline{I}$, and removing all edges from the group U ti these items. The explanation in this case is a set of items, rather than a set of edges. For an item $i \in \overline{I}$, we approximate the effect of its removal from group U by computing $\Delta(U, I, i) = \sum_{u \in U} \Delta(U, I, (u, i))$. We sort the items according to the Δ-values, and return the top ones that achieve the target bias goal. We will refer to this algorithm as ItemExplain.

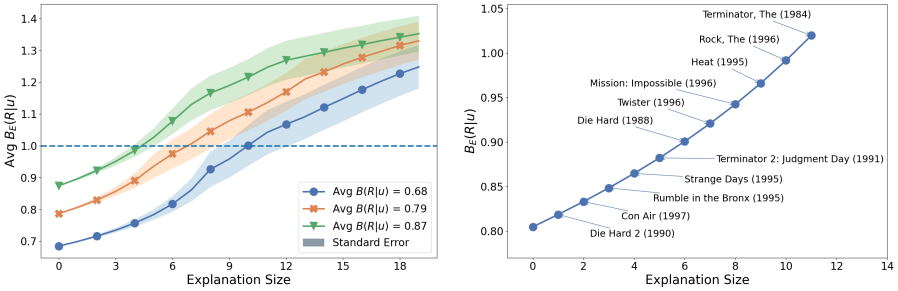
Item Group Explanations. In the case of item group explanations, we want to explain why for a group of items I, the recommendation algorithm estimates lower scores from group U than $\overline{U}$. For this case, we adopt the three different types of explanations we described for the user group explanations, and the corresponding algorithms.

4 Experimental Evaluation

We now evaluate our algorithms for producing explanations for the different recommendation biases. The goal of the experiments is to understand quantitatively and qualitatively the different explanations we produce. Code and data are available in our online github repository.

4.1 Datasets

We evaluate our algorithms using both real and synthetic datasets. We will now describe our datasets and their characteristics.



(a) Greedy explanations (b) Explanations Example

Fig. 1: Individual user explanations

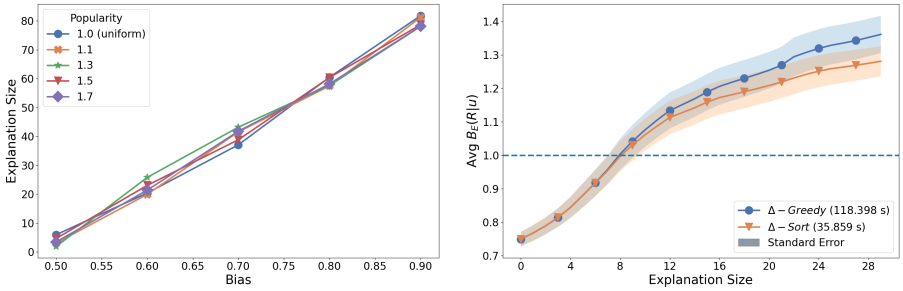
Real Dataset: We use the MovieLens 100K Dataset [8] which is a dataset of user ratings on movies. It consists of 100,000 ratings from 943 users on 1,682 movies. There is demographic information about the users, and genre information about the movies. We used the gender to define groups of users, and the movie genre to define groups of movies. We used a subset of the dataset with the movies from the Action and Romance genres. The resulting dataset consists of all users, 670 males and 273 females and 448 movies, 226 Action and 222 Romance. In our experiments we will denote the group of Males as M and the group of Females as F, and the Romance group as R and the Action group as A.

Synthetic Datasets: We also created synthetic datasets to study our algorithms. Our datasets consist of $N_U = 1000$ users and $N_I = 1000$ items. Users are partitioned into two groups, U_0 and U_1, of equal size, and items into two categories, I_0 and I_1, also of equal size. We allocated 100 ratings per user (10% of the items). We introduced bias in the data, where users in U_0 favor items in I_0, and users in U_1 favor items in I_1. We control the bias with a parameter β: For a user from group U_0 (U_1), with probability β we generate a rating to an item in category I_0 (I_1), and with probability $1-\beta$ to an item in category I_1 (I_0). In our allocation of ratings we ensure that each item has at least 5 ratings. The goal is to study the effect of the data bias in the explanations, so we vary the parameter β to take values in $\{0.5, 0.6, 0.7, 0.8, 0.9\}$.

We also want to investigate the effect of item popularity, so we varied the probability distribution with which we select an item within an item category. We generated popularity distributions for the items utilizing Zipf's Law. For the Zipf law parameter a we used parameters $a \in \{1, 1.1, 1.3, 1.5, 1.7\}$, where higher parameter value, implies more skewed distribution, while value $a = 1$ results in a uniform distribution.

4.2 Individual user bias explanations

We first experiment with individual user bias explanations. For the following we will use B instead of B_{R_D} for the preference ratio, and B_E instead of $B_{R_D|E}$ to



(a) Synthetic Data Explanations (b) Greedy vs Sort

Fig. 2: Individual user explanations: Synthetic Data and Comparisons

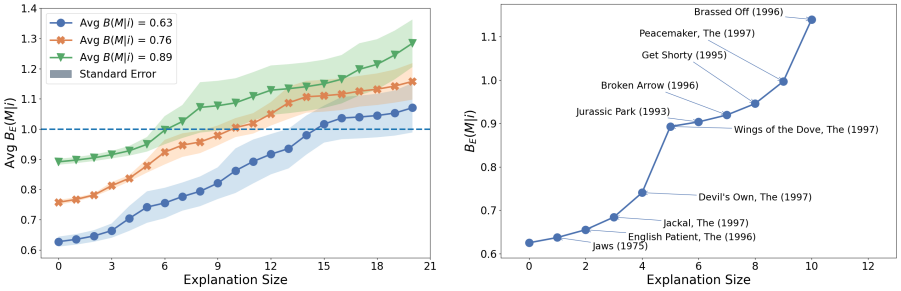
denote the preference ratio when applying explanation change E. In the Movies dataset, we select the target item group I to be the movies in the Romance category (group R). We consider users with initial $B(R|u) < 1$, and we seek explanations that will result in $B_E(R|u) \geq 1$ (target $\theta = 1$).

To study the how the initial bias affects the complexity of our explanations, we sampled 20 users in three different ranges of initial $B(R|u)$ values: $(0.65, 0.75)$, $(0.75, 0.85)$, and $(0.75, 0.85)$. Figure 1a plots for the Greedy algorithm the size of the explanation E_u and the resulting (average) preference ratio $B_{E_u}(R|u)$. We observe that the more biased the nodes initially in favor of Action and against Romance (lower $B(R|u)$), the larger the complexity of the explanation. However, even for strong anti-Romance bias, we can explain it with a small number of edges (no more than 12 for $B(R|u) \approx 0.7$ and less than 5 for $B(R|u) \approx 0.9$).

An example of the explanation movies is shown in Figure 1b, where we plot $B_E(R|u)$ over the iterations of the Greedy algorithm, and the movie selected at each iteration. The selected movies are all well-known Action movies, such as "Die Hard", "Mission Impossible" and "Terminator".

We performed some additional measurements in order to better understand the type of movies that the algorithm selects as explanations, and more specifically how the popularity of the movie affects the selection. For a user u, we compute the correlation between the Δ-values, $\Delta(u, R, (u, i))$, of the selected movies and their (degree) popularity (for more details see the Supplementary Material). We observe a clear negative correlation, meaning that the movies selected have small degree. It is thus the case that removing edges to fringe movies has a stronger effect than removing edges to popular movies. This can also be deduced from Equation 2, where the target of the selected edge must have lower probability of reaching the target group, than the other neighbors of the user. Unpopular Action movies are less likely to lead to the Romance category.

We also perform experiments with synthetic data, shown in Figure 2a. We vary the bias β in the data, as well as the skewness of the degree distribution a. We observe that as the bias increases the complexity of the explanations increases. The skewness of the distribution does not seem to affect the explana-



(a) Explanation size (b) Oscar & Lucinda explanations

Fig. 3: Individual item Explanations

tion size. In the synthetic data we observe an even stronger negative correlation between the Δ-value and the popularity of the movie.

Finally, we perform a comparison of the GREEDY algorithm with the heuristic algorithm SORT, in terms of explanation size and complexity, shown in Figure 2b. We observe that we gain considerably in efficiency ($\sim$120 sec for the GREEDY algorithm, while $\sim$36 sec for the SORT algorithm). At the same time we incur only a small increase in the explanation complexity for large values of target value θ. Thus, the sorting algorithm is a viable alternative to the GREEDY solution.

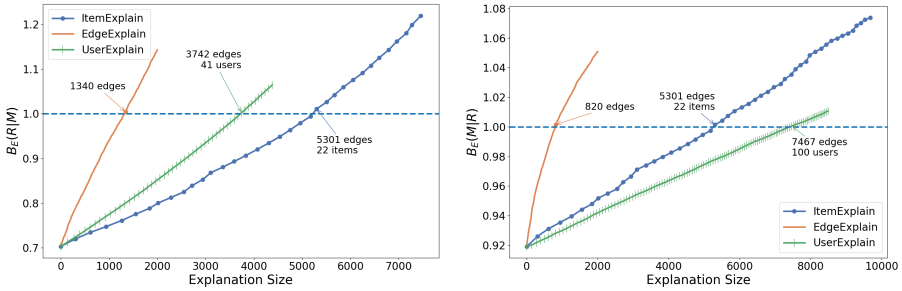
4.3 Individual item bias explanations

We now consider individual item bias explanations. In the Movies dataset we set the target user group U to be the group of male users M, and the target value $\theta = 1$. We consider movies with initial $B(M|i) < 1$, and we create again samples of 20 movies, for three different ranges of initial $B(M|i)$ values: $(0.55, 0.65)$, $(0.7, 0.8)$, and $(0.85, 0.95)$. Figure 3a plots the size of the explanation E_i and the resulting (average) $B_{E_i}(M|i)$. We observe that the stronger the bias against Males (lower $B(M|i)$), the larger the complexity of the explanation. However, we can still explain the bias with a small number of edges (15 on average for initial $B(M|i) \approx 0.6$, and 6 on average for initial $B(R|u) \approx 0.9$).

An example explanation for the movie "Oscar & Lucinda" is shown in Figure 3b. The selected movies contain known Action movies like "Jaws" or "The Jackal", but also Romance movies like "English Patient" or "The Wings of the Dove". This is because the algorithm prioritizes the selection of movies from users with low number of total ratings, since the removal of edges from such users has a greater effect. Also, the movies selected are popular; removing edges to them helps the random walk to allocate more probability to the selected item.

4.4 User group bias explanations

We now consider user group bias explanations. In the Movies dataset we set the target item group to be the Romance category R, and the target user to



(a) User Group Explanations (b) Item Group Explanations

Fig. 4: Group explanations

be the male users M. We have that $B(R|M) \approx 0.70$, so there is a bias of the recommender against the Romance category when producing scores for the Male users, which we want to explain. We set bias target value $\theta = 1$.

We consider the three different algorithms for producing explanations that we described in Section 3.3: EdgeExplain, UserExplain, ItemExplain. Each algorithm produces a different type of explanations. We plot them together in Figure 4a. The x-axis shows the number of edges selected and the y-axis $B(R|M)$. On the plot we also show the number of users selected for UserExplain, and the number of items selected for ItemExplain, to achieve the target value.

We observe that in terms of edges removed the EdgeExplain algorithm has the most efficient explanation, followed by the UserExplain algorithm, and then the ItemExplain algorithm. This is expected since the last two produce a different kind of explanations. These explanations are interesting in their own right. The UserExplain algorithm can explain the bias by affecting only a small subset of 44 users, while the ItemExplain algorithm produces an explanation with just 22 movies. Looking at the selected edges of EdgeExplain, we observed that they involve 320 distinct users, and 148 distinct movies. The occurrence frequency histogram for the users indicates that we never remove a lot of edges from a user. On the other hand, the histogram for movies is more skewed, with many movies appearing a few times, and a few movies having several edges removed. The frequency histograms appear in the Supplementary Material.

Looking into the characteristics of the explanations, the EdgeExplain algorithm tends to select edges from users with low degree and few edges towards Action. Removing such edges has a strong effect, as it transfers significant amount of probability to the Romance group. This is in contrast to the UserExplain algorithm that selects users with high degree, and several ratings in Action. Removing all of these edges results in high increase of $B(R|M)$. There is zero overlap between the users affected by the two algorithms.

The ItemExplain algorithm tends to select movies that are overall popular. The 22 movies selected by ItemExplain appear also in the top-100 edges selected by EdgeExplain, while 16 out of the 22 selected movies appear in

the top-100 most popular movies. The top-5 selections of the ItemExplain algorithm are: "Air Force One", "The Godfather", "The Princess Bride", "Independence Day", "Star Treck: First Contact". We see that the list contains popular movies that are also popular outside of the Action genre, such as "The Godfather" or "The Princess Bride".

We also experimented with synthetic data. Results appear in the Supplementary Material. We observe again that as the bias increases the explanation complexity also increases, while increasing the skewness of the degree distribution (high a values) results in higher explanation size.

4.5 Item group bias explanations

Finally, we consider item group bias explanations. We set again the target user group U to be the group of male users M, and the target item group to be the Romance category R. We have initial $B(M|R) \approx 0.92$, so there is bias in the recommender against Males when producing scores for Romance, which we want to explain. We set $\theta = 1$.

We consider again the three different algorithms for producing explanations for groups: EdgeExplain, UserExplain, ItemExplain. Note that the selections of the algorithms are exactly the same as for the user-group case. The resulting behavior though is different, as shown in Figure 4.

We first observe that the EdgeExplain achieves the target value much faster, and with a steep increase. The ItemExplain algorithm coincidentally achieves the target bias value at the same number of movies as for the user-group case. The algorithm in this case is better than the UserExplain algorithm, which performs much worse, both in terms of number of edges removed and number of users affected. The UserExplain algorithm selects users with many edges to the Action category. This increases $R_{D|E}(M, R)$, but it also increases $R_{D|E}(F, R)$, so the increase in $B_E(M|R)$ is small. This is in contrast with the EdgeExplain algorithm that selects edges from users with small degree, that cause large increase to $R_{D|E}(M, R)$ but small increase to $R_{D|E}(F, R)$.

5 Related Work

The problem of bias and fairness in recommendations has received a lot of recent attention [10, 22]. A general distinction is on whether fairness is considered at the level of individuals or groups [4]. In recommendations in particular, there are also many sides involved, as fairness can be examined from both the consumer and producer perspective [3]. In this paper, we provide explanations both at the individual and group level as well as both for the user and the item side.

Various perspectives of fairness have been considered. One perspective is that the rating prediction errors must be similar across groups or individuals [23, 11]. Accuracy based fairness is also formulated using pairwise metrics [1]. Another perspective is fair exposure, for example, allocating exposure to items in recommendation lists proportional to their relevance [14, 2]. Finally, calibration

asks that the predicted proportions of the recommended items, or groups agree with the corresponding proportions in the user preferences [19, 15]. In this paper, we offer a general method for explaining unfair behavior and applied it to explain cases where there are discrepancies between predictions and group proportions.

Explainability in AI is getting increasing attention. It is achieved by using interpretable and transparent models, or by generating post-hoc explanations for opaque models. A common approach in the latter case are attribution-based methods, including methods that quantify how much the output is changed when an input variable is perturbed, and methods that quantify marginal effects of variables on the output compared to a reference model [16]. Well-known examples of such methods are LIME [12] and DeepLIFT [13]. As opposed to attribution techniques, counterfactual explanations produce small changes in the input so that a different prediction is made [21]. In this paper, we take a post-hoc countefactual-based approach.

There has been much work on explaining recommendation results [24]. Counterfactual explanations for recommendations explore either item features, or user actions. An example of the former is CountER that formulates an optimization problem to generate minimal changes on the features of an item such that the recommendation decision about the item is reversed [17]. Prince [7] follows the latter approach for graph recommenders and looks for a set of minimal user actions that, if removed, the top recommendation item will be replaced by a different item. ACCENT extends the user action approach to neural recommenders [18]. Our work extends the user action approach for explaining unfairness. The only other work on counterfactual explanations for unfairness in recommendations that we are aware of is [6] that follows an item feature approach.

Finally, there is a line work on graph perturbations to achieve specific properties. For example, previous research has studied the addition of edges via link recommendations for increasing the Pagerank of underrepresented groups [20], while rewiring edges was proposed to decrease paths to polarized content [5].

6 Conlusions

In this work we considered the problem of defining counterfactual explanations for bias of recommendation algorithms. We considered different types of bias, and provided definitions for the explanations for these biases. We studied the case of a random walk recommender, and we provided efficient algorithms for computing different types of explanations. We validated our approach with experiments on real and synthetic data. For future work, we are interested in considering other definitions of bias and fairness, and extending our approach to more recommendation algorithms.

Acknowledgements: This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

References

1. Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., Heldt, L., Zhao, Z., Hong, L., Chi, E.H., Goodrow, C.: Fairness in recommendation ranking through pairwise comparisons. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019. pp. 2212–2220. ACM (2019)
2. Biega, A.J., Gummadi, K.P., Weikum, G.: Equity of attention: Amortizing individual fairness in rankings. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018. pp. 405–414. ACM (2018)
3. Burke, R.: Multisided fairness for recommendation. CoRR **abs/1707.00093** (2017), http://arxiv.org/abs/1707.00093
4. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.S.: Fairness through awareness. In: Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012. pp. 214–226. ACM (2012)
5. Fabbri, F., Wang, Y., Bonchi, F., Castillo, C., Mathioudakis, M.: Rewiring what-to-watch-next recommendations to reduce radicalization pathways. In: WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022. pp. 2719–2728. ACM (2022)
6. Ge, Y., Tan, J., Zhu, Y., Xia, Y., Luo, J., Liu, S., Fu, Z., Geng, S., Li, Z., Zhang, Y.: Explainable fairness in recommendation. In: SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022. pp. 681–691. ACM (2022)
7. Ghazimatin, A., Balalau, O., Roy, R.S., Weikum, G.: PRINCE: provider-side interpretability with counterfactual explanations in recommender systems. In: Caverlee, J., Hu, X.B., Lalmas, M., Wang, W. (eds.) WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020. pp. 196–204. ACM (2020)
8. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4) (dec 2015). https://doi.org/10.1145/2827872, https://doi.org/10.1145/2827872
9. Nikolakopoulos, A.N., Karypis, G.: Recwalk: Nearly uncoupled random walks for top-n recommendation. In: Culpepper, J.S., Moffat, A., Bennett, P.N., Lerman, K. (eds.) Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019. pp. 150–158. ACM (2019)
10. Pitoura, E., Stefanidis, K., Koutrika, G.: Fairness in rankings and recommendations: an overview. VLDB J. **31**(3), 431–458 (2022)
11. Rastegarpanah, B., Gummadi, K.P., Crovella, M.: Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019. pp. 231–239. ACM (2019)
12. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144. ACM (2016)
13. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Precup, D., Teh, Y.W. (eds.) Proceedings

of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 3145–3153. PMLR (2017)
14. Singh, A., Joachims, T.: Fairness of exposure in rankings. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018. pp. 2219–2228. ACM (2018)
15. Steck, H.: Calibrated recommendations. In: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018. pp. 154–162. ACM (2018)
16. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 3319–3328. PMLR (2017)
17. Tan, J., Xu, S., Ge, Y., Li, Y., Chen, X., Zhang, Y.: Counterfactual explainable recommendation. In: CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021. pp. 1784–1793. ACM (2021)
18. Tran, K.H., Ghazimatin, A., Roy, R.S.: Counterfactual explanations for neural recommenders. In: SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021. pp. 1627–1631. ACM (2021)
19. Tsintzou, V., Pitoura, E., Tsaparas, P.: Bias disparity in recommendation systems. In: Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019. CEUR Workshop Proceedings, vol. 2440. CEUR-WS.org (2019)
20. Tsioutsiouliklis, S., Pitoura, E., Semertzidis, K., Tsaparas, P.: Link recommendations for pagerank fairness. In: WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022. pp. 3541–3551. ACM (2022)
21. Verma, S., Dickerson, J.P., Hines, K.: Counterfactual explanations for machine learning: A review. CoRR **abs/2010.10596** (2020), https://arxiv.org/abs/2010.10596
22. Wang, Y., Ma, W., Zhang, M., Liu, Y., Ma, S.: A survey on the fairness of recommender systems. CoRR **abs/2206.03761** (2022)
23. Yao, S., Huang, B.: Beyond parity: Fairness objectives for collaborative filtering. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 2921–2930 (2017)
24. Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. Found. Trends Inf. Retr. **14**(1), 1–101 (2020)