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

FaiRecSys: Mitigating Algorithmic Bias in Recommender Systems

**Bora Edizel · Francesco Bonchi · Sara Hajian ·
André Panisson · Tamir Tassa**

Received: date / Accepted: date

Abstract Recommendation and personalization are useful technologies which influence more and more our daily decisions. However, as we show empirically in this paper, the bias that exists in the real world and which is reflected in the training data, can be modeled and amplified by recommender systems, and in the end returned as biased recommendations to the users. This feedback process creates a self-perpetuating loop which progressively strengthens the filter bubbles we live in. Biased recommendations can also reinforce stereotypes such as those based on gender or ethnicity, possibly resulting in disparate impact.

In this paper we address the problem of algorithmic bias in recommender systems. In particular, we highlight the connection between *predictability of sensitive features* and bias in the results of recommendations and we then offer a theoretically founded bound on recommendation bias based on that connection. We continue to formalize a fairness constraint and the price that one has to pay, in terms of alterations in the recommendation matrix, in order to achieve fair recommendations. Finally, we propose FaiRecSys – an algorithm that mitigates algorithmic bias by post-processing the recommendation matrix with minimum impact on the utility of recommendations provided to the end-users.

Keywords: Algorithmic bias; Recommender systems; Fairness; Privacy

1 Introduction

Recommender systems are nowadays a pervasive technology influencing our daily lives and strengthening the *filter bubbles* in which we all live: the media we consume, the stories we read, the people we connect to, the places we visit, the jobs we apply to, and the ads we see on the Web. It is therefore of societal and ethical importance to ask whether collaborative filtering algorithms, used for recommendation and personalization, might be involuntarily perpetuating existing bias towards some specific demographic groups. It turns out that the answer is positive: for instance, recent studies have shown that Google's online advertising system displays ads for high-income jobs to men much more often than it does to women [4]; and ads related to arrest records are significantly more likely to show up on searches for distinctively black names or a historically black fraternity [32].

Note that this *algorithmic bias* [11] exists even when there is no discrimination intention in the developer of the algorithm, and even when the recommender system does not take as input any demographic information: nevertheless, by carefully exploiting items' and users' similarities, the algorithm might end up recommending an item to a very homogeneous set of users. For instance, the algorithm would be considered biased if the set of users to which it offers a book entitled "How to

Bora Edizel
Pompeu Fabra University, Barcelona, Spain.
E-mail: bora.edizel@upf.edu

Francesco Bonchi
ISI Foundation, Torino, Italy.
E-mail: francesco.bonchi@isi.it

Sara Hajian
Eurecat, Barcelona, Spain.
E-mail: sara.hajian@eurecat.org

André Panisson
ISI Foundation, Torino, Italy.
E-mail: andre.panisson@isi.it

Tamir Tassa
The Open University, Ra'anana, Israel.
E-mail: tamirta@openu.ac.il

be a leader in the hi-tech industry" would include too few women, or if a movie about black gangs in west LA will be offered mainly to black users. Even when recommendations are of high quality, they might be not well perceived by the user that might find them "too accurate" and discriminatory.

Methods for removing statistical bias or modeling such bias in order to improve the performance of recommender systems is an important topic in the area. Many methods have been proposed to leverage algorithmic bias to improve recommendation accuracy, sometimes amplifying the bias already present in the data. The ability to identify systematic tendencies for users to give higher ratings than others, or users that change their baseline ratings over time, has a big impact on the performance of recommender systems [22]. Some of these statistical biases (e.g. temporal bias) are part of the dynamics of interest in recommended items and must be modeled accordingly.

In this paper we address the problem of algorithmic bias that might lead to discriminatory behavior in recommender systems. The methods presented in this work go in the opposite direction to that of other methods, such as the ones presented in [22]. While our proposed methods are not intended to increase the performance of recommender systems, we show how to reduce the bias, and how to do that in a manner that minimizes the inevitable resulting reduction in performance.

Before presenting our contributions, we provide a motivating empirical example on real-world data.

1.1 Motivating empirical example

The effects of algorithmic bias can be better understood through real-world examples. For this, we built a recommender system using data collected from the **Reddit** website[1]. (In Section 5 we provide more details about the dataset collection.) **Reddit** is an entertainment, social networking, and news website where registered members can submit stories (either as text posts or links), making it essentially an online bulletin board system. Registered users can then vote submissions up or down to determine their position on the site's pages. Content entries are organised by areas of interest called "subreddits". Subreddit topics include news, politics, gaming, movies, music, books, and many others. Users can also comment the submissions, and respond back and forth in a conversation-tree of comments. Gender attributes are not supported by the **Reddit** platform. However, in some subreddits users can report their gender as part of the subreddit rules. We selected a set U of users that reported their gender and submitted comments in a set I of subreddits. In this exercise, a user commenting in a subreddit is interpreted as an implicit positive feedback. We then built a binary matrix $R = (r_{u,i} : u \in U, i \in I)$ where $r_{u,i}$ is set to 1 if user u posted a comment in subreddit i, and 0 otherwise. We then used R to train a *weighted regularized matrix factorization* (WRMF) recommendation model [16], which is appropriate for implicit feedbacks. The model parameters were selected using 10-fold cross-validation. The output is a binary matrix $C = (c_{u,i} : u \in U, i \in I)$ with the top-10 recommendations given by the system; specifically, $c_{u,i}$ is set to 1 if subreddit i is among the top-10 subreddits recommended to user u, and u had not posted thus far a comment in that subreddit.

The effects of algorithmic bias can be seen in the resulting recommendation matrix C. For example, the subreddit "MakeupAddiction" is very popular among females - in the **Reddit** dataset (the matrix R), 90% of the users who submitted a comment to this subreddit were females. When producing recommendations, the generated model reinforces the imbalance between males and females, by assigning 97% of this subreddit recommendations to females. Another example is the subreddit "cscareerquestions", where users discuss Computer Science careers. This subreddit is more popular among males than among females: 84% of the users who posted comments in this subreddit were males, and only 16% were females. We found that the WRMF model reinforces this bias, producing 90% of this item's recommendations to males, and only 10% to females. In fact, although gender is never seen by the algorithm, latent variables that are related to gender might be produced by combining the entries of the matrix R, and the generated model reinforces the imbalance between males and females. In our example, we found that 95% of subreddits that are popular among females show imbalance reinforcement, while 87% of subreddits popular among males have imbalance reinforced towards males.

The above bias reinforcement effect might be specific to the model produced in this example, but the methodology presented in this work can help to mitigate the biases in an effective way regardless of the model being used. Our goal is to reassign recommended items to users by minimizing the predictability of their gender (or any other sensitive attribute) from the recommendations while preserving the utility of recommendations as high as possible. This should help in reducing the imbalance between males and females, but it should also usher in more desirable outcomes by encouraging female users to participate in discussions that are typically dominated by male users, and vice-versa.

[1] `https://www.reddit.com/`

1.2 Paper contributions and roadmap

As we have shown in the above example, the bias existing in the real world (and thus in the training data), can be strengthened by recommender systems, in a self-perpetuating loop [11] if not treated appropriately. The problem of algorithmic bias and the possibility that decisions informed by data mining algorithms may have discriminatory effects, even in the absence of discriminatory intent, have received recently a great deal of attention.[2] However, most of the technical efforts thus far in addressing those issues focused on predictive tasks (literature is surveyed later in Section 2).

In this paper we address the problem of algorithmic bias in recommender systems. Consider the user-item recommendation matrix that the recommendation algorithm outputs; there is a direct connection between bias in the recommendation matrix, based on some sensitive attribute (say, gender), and the predictability of that sensitive attribute from the recommendation matrix. Stated differently, if by simply looking at the rows of the recommendation matrix (which hold the recommendations for each user) it is possible to predict the user's gender with high accuracy, such predictability indicates a high gender-based bias in the recommendations. Hence, in order to mitigate such a bias, our goal is to limit the *predictability of sensitive features* from the recommendation matrix. A recommendation matrix is then considered to be ε-fair, if it is impossible to predict from it the users' sensitive attributes to within an average class-conditioned error smaller than ε. (We focus in this study on binary sensitive attributes, and use the term gender as our running example for the sake of simplicity and clarity.)

In order to achieve ε-fairness with respect to a given sensitive attribute, for some preset value of ε, it is needed to modify the entries of the recommendation matrix. The price of ε-fairness is then defined as the distance between the original recommendation matrix C and the closest matrix C' which complies with the ε-fairness constraint. That price indicates the minimal number of alterations that must be introduced in C until it becomes ε-fair. Finally, we propose FaiRecSys – an algorithm that mitigates algorithmic bias with respect to a given sensitive attribute, by post-processing a recommendation matrix with minimum impact on the utility of recommendations provided to the end-users.

It should be noted that incorporating this fairness constraint introduces a tradeoff, as the fairness constraint is essentially contradicting utility. It is worth noticing that this situation essentially exists in any prediction task, where anti-discrimination, privacy or fairness constraints contradict accuracy. This is also typical in all privacy-preserving techniques. Higher protection of privacy can be achieved only by higher deteriorations of utility. Hence, the usual practice is to determine the desired level of privacy-preservation and then find a solution which meets that privacy goal while maintaining the highest possible utility. For example, in the context of k-anonymity, given an input table T, and an anonymity threshold k, the goal is to find another table $T' = A(T)$ (where A is the anonymization algorithm) where T' is k-anonymous and the distance between T and T' is minimal. The problem that we consider here is similar: given a recommendation matrix C and a fairness threshold ε, we wish to compute C' that is ε-fair and has a minimal distance to the original C. The matrix C' is not "bias-free" or "balanced". But, depending on the input ε, it is less biased than C.

The rest of the paper is organized as follow. In the next section we briefly survey related literature. In Section 3 we formalize the problem studied while in the following section we provide a characterization of the fairness property that we wish to achieve, leading to the proposed algorithm. Finally, Section 5 presents our experimental analysis.

For the sake of readability, the proofs of all our mathematical claims are given in Appendix A.

2 Related work

The problem of algorithmic bias and discrimination has received recently a lot of attention in the data mining community [11]: some effort has been devoted to the problem of detecting and measuring existing discrimination in the data [29,28,31,25,26,30]; while other proposals [19,18,12,13,14,5,33] aimed at ensuring that data mining models do not lead to discriminatory decisions even if the training dataset is inherently biased. The bulk of this literature focuses on the classification task. While our work also lies in the area of bias prevention, it focuses on recommender systems.

Bias prevention approaches can be classified according to the phase of the data mining process in which they operate: *pre-processing*, *in-processing* and *post-processing* methods [11]. Pre-processing methods aim to control distortion of the training set. In particular, they transform the training dataset in such a way that the discriminatory biases contained in the dataset are smoothed, hampering the mining of unfair decision models from the transformed data. In-processing methods modify recommendation algorithms in such a

[2] "Big Data: Seizing Opportunities, Preserving Values." US Executive Office of the President, May 2014. https://obamawhitehouse.archives.gov/sites/default/files/docs/big_data_privacy_report_may_1_2014.pdf

way that the resulting models do not entail unfair decisions. Lastly, post-processing methods act on the extracted data mining model results, instead of acting on the training data or on the algorithm. The method presented in this work focuses on the post-processing phase, correcting potential biases in the output of the recommendation algorithm.

A large number of studies focused on fairness in classification, e.g., preventing discrimination against individuals based on their membership in some group, or fairness in prediction of sensitive attributes in order to avoid issuing decisions based on those attributes. Many works involving post-processing methods [28,3,17,5] also focused on prediction tasks. Feldman et al. [7] defined the notion of fairness based on predictability of sensitive attributes. We used a similar notion of fairness, but applied it in the context of recommendation systems. Another difference between the present work and [7] is that we offer post-processing methods while [7] focused on pre-processing methods.

In the literature that includes recommender systems, Kamishima et al. [20] defined the notion of "recommendation independence" which imposes statistical independence between recommendation results and sensitive attributes. They proposed an in-processing method that injects a regularization term into the objective function of matrix factorization in order to enhance independence between recommendation results and sensitive attributes. This statistical independence can be enhanced by probabilistic generative models [21] and can be useful to prevent bias, but it does not offer a theoretical bound for the bias in recommendation results. E.g., in a more practical setting, it might be a problem for a company to argue that their recommendations are not biased using measures based on statistical independence, whilst our approach offers a theoretically founded bound on recommendation bias based on predictability of sensitive attributes. Moreover, such in-processing methods are not suitable for use after the recommendations have been generated, whereas our proposed method acts in a post-processing phase and can correct bias after recommendations have been produced by the recommendation algorithm.

Recently, Ekstrand et al. [6] show that recommender systems suffer of what is known as "sample size bias", exactly as it is the case for supervised machine learning. This is to say that recommender systems perform better for the dominant subgroup, even when the subgroup feature is not used. In particular, Ekstrand et al. show empirically that popular recommendation algorithms work better for males since the majority of the users in the dataset are males.

Burke [2] presents a taxonomy of classes for fair recommendation systems. He mentions 3 sides: users, providers and platform. He argues that different recommendation settings should have different fairness requirements such as "fairness for only users", "fairness for only providers" and "fairness for both users and providers". Considering that taxonomy, our work falls into "fairness for only users" category where there are only users and system in the setting. No concrete method is proposed.

3 Problem statement

Let $U = \{u_1, \ldots, u_N\}$ be the set of users to whom recommendations are issued, and let $I = \{i_1, \ldots, i_M\}$ be the set of items. The output of recommendation systems is typically a matrix as follows:

Definition 1 A *recommendation matrix* is an $N \times M$ binary matrix $C \in \{0,1\}^{U \times I}$. Namely, it is a matrix that associates a binary value to any pairing of a user $u \in U$ and an item $i \in I$. Specifically, $C(u,i) = 1$ means that item i is recommended to user u, and $C(u,i) = 0$ means that it is not.

Specifically, we will be interested in recommendation matrices in which each user is recommended the same number of items k:

Definition 2 Let k be a fixed integer $0 < k \ll M$; then C is called k-weighted if for all $u \in U$, $\sum_{i \in I} C(u,i) = k$ (namely, if every user is recommended k items).

Assume that each user has a (binary) sensitive attribute and let $\mathbf{b} = (b(u) : u \in U) \in \{0,1\}^U$ be the vector where $b(u)$ is u's sensitive attribute. The recommendation matrix C could be used to predict the users' sensitive attributes.

Definition 3 Let $f : \{0,1\}^I \to \{0,1\}$ be a function that, given $C(u,\cdot)$, predicts $b(u)$, $u \in U$. Then f is called a *predictor* for $\mathbf{b}$, and $\mathbf{f} = \mathbf{f}(C) := (f(C(u,\cdot)) : u \in U)$ is a prediction of $\mathbf{b}$.

Given a predictor, we proceed to define its prediction error.

Definition 4 Let f be a predictor for $\mathbf{b}$ and $\mathbf{f}$ be the corresponding prediction. Given $p \neq q \in \{0,1\}$, we let

$$Pr[\mathbf{f} = p | \mathbf{b} = q] = \frac{|\{u \in U : f(C(u,\cdot)) = p \wedge b(u) = q\}|}{|\{u \in U : b(u) = q\}|}$$

denote the relative number of users for which the sensitive value equals q but the prediction wrongly predicted it to be $p \neq q$. Then f's *Balanced Error Rate (BER)* is $\mathbf{f}$'s average class-conditioned error:

$$BER(\mathbf{f}, \mathbf{b}) = \frac{Pr[\mathbf{f} = 1|\mathbf{b} = 0] + Pr[\mathbf{f} = 0|\mathbf{b} = 1]}{2} . \quad (1)$$

Those definitions give rise to the notions of predictability and fairness.

Definition 5 $\mathbf{b}$ is said to be *ε-predictable* by C if there exists a predictor f for which $BER(\mathbf{f}, \mathbf{b}) \leq \varepsilon$. A recommendation matrix C is said to be *ε-fair* with respect to $\mathbf{b}$ if $\mathbf{b}$ is not ε-predictable from C.

Note that a constant predictor, $f \equiv 0$ or $f \equiv 1$, has a BER of 1/2. Therefore, every sensitive vector $\mathbf{b}$ is ε-predictable for every $\varepsilon \geq 1/2$. Consequently, a recommendation matrix C can be ε-fair only if $\varepsilon < 1/2$.

Next, we formalize our problem. Given a fairness threshold $\varepsilon \in (0, 1/2)$, a recommendation matrix C, and a sensitive vector $\mathbf{b}$, we wish to find *an alternative* recommendation matrix $C' \in \{0,1\}^{U \times I}$ which is ε-fair and minimizes some distance $\mathrm{dist}(C, C')$ from the original C. Let μ be a metric on the space of recommendation vectors $\{0,1\}^I$, and let

$$\mathrm{dist}(C, C') := \sum_{u \in U} \mu(C(u, \cdot), C'(u, \cdot)) . \quad (2)$$

be the induced distance function on the set $\{0,1\}^{U \times I}$ of recommendation matrices over U and I. Then the price of fairness is defined as follows.

Definition 6 For a k-weighted recommendation matrix $C \in \{0,1\}^{U \times I}$, $\mathbf{b} \in \{0,1\}^U$, and $\varepsilon < 1/2$, let $\Gamma_{C,\mathbf{b},\varepsilon}$ be the set of all k-weighted matrices $C' \in \{0,1\}^{U \times I}$ that are ε-fair with respect to $\mathbf{b}$. Then *the price of ε-fairness* for C is $\min_{C' \in \Gamma} \mathrm{dist}(C, C')$.

Namely, given a definition of distance between matrices, the price of ε-fairness for C is defined as the distance between C and the closest matrix C' which is ε-fair with respect to $\mathbf{b}$. We are now ready to define the relevant computational problem in this context:

Problem 1 (Problem FRM – Fair Recommendation Matrix) Given $C \in \{0,1\}^{U \times I}$, $\mathbf{b} \in \{0,1\}^U$, and $\varepsilon < 1/2$, find $C' \in \Gamma_{C,\mathbf{b},\varepsilon}$ that minimizes $\mathrm{dist}(C, C')$.

We consider two metrics on $\{0,1\}^I$:

1. $\mu_1(\mathbf{y}, \mathbf{z}) := \begin{cases} 0 & \mathbf{y} = \mathbf{z} \\ 1 & \mathbf{y} \neq \mathbf{z} \end{cases}$; such a metric induces a distance function between matrices that counts the number of users that are affected (in any way) by replacing C with C'; and
2. $\mu_2(\mathbf{y}, \mathbf{z}) = \frac{1}{2} \cdot \|\mathbf{y} - \mathbf{z}\|_1$; such a metric induces a distance function between matrices that counts the number of item recommendations that are changed when switching from C to C'.

4 Analysis and algorithms

4.1 Characterizing ε-fairness

Here we offer a simple characterization of ε-fair matrices. For the sake of convenience, we refer hereinafter to users for whom $b(u) = 0$ as "men" and users for whom $b(u) = 1$ as "women", and denote their numbers by b_0 and $b_1 = N - b_0$ respectively.

Definition 7 Let $Y = \{\mathbf{y} \in \{0,1\}^I : \exists u \in U$, such that $C(u, \cdot) = \mathbf{y}\}$. Then the corresponding *contingency table* H is a matrix of 2 rows and $|Y|$ columns where for each $x \in \{0,1\}$ and $\mathbf{y} \in Y$, $H(x, \mathbf{y})$ equals the number of users $u \in U$ for which $b(u) = x$ and $C(u, \cdot) = \mathbf{y}$.

Any contingency table H induces a predictor as we define next.

Definition 8 *H induces a predictor $f_H : Y \to \{0,1\}$ where, for each $\mathbf{y} \in Y$, $f_H(\mathbf{y}) = 0$ if $\frac{H(0,\mathbf{y})}{b_0} \geq \frac{H(1,\mathbf{y})}{b_1}$ and $f_H(\mathbf{y}) = 1$ otherwise; f_H is called the Memory-Based Predictor (MBP).*

Namely, given a vector of recommendations $\mathbf{y} \in \{0,1\}^I$, the MBP f_H returns the "gender" in which that recommendation vector has a larger relative frequency (where in case of a tie, f_H favors the gender "male").

The next theorem spells out the property of f_H that grants it a special place in our discussion.

Theorem 1 *Let f_H be the MBP and $f \in \Pi$ be any other predictor. Denote by $\mathbf{f}_H = (f_H(C(u, \cdot)) : u \in U)$ and $\mathbf{f} = (f(C(u, \cdot)) : u \in U)$ the corresponding prediction vectors. Then $BER(\mathbf{f}_H, \mathbf{b}) \leq BER(\mathbf{f}, \mathbf{b})$.*

(The reader is reminded that, for the sake of readability, all proofs are given in Appendix A.)

Theorem 1 implies that C is ε-fair with respect to $\mathbf{b}$ iff $BER(\mathbf{f}_H, \mathbf{b}) > \varepsilon$. Namely, in order to test ε-fairness it suffices to examine a single predictor only — the MBP.

We are now ready to spell out an explicit algebraic condition of ε-fairness, which can be easily verified by examining the contingency table.

Theorem 2 *Define*

$$\beta_H := \frac{\sum_{\mathbf{y}: \frac{H(0,\mathbf{y})}{b_0} < \frac{H(1,\mathbf{y})}{b_1}} H(0, \mathbf{y})}{b_0} + \frac{\sum_{\mathbf{y}: \frac{H(0,\mathbf{y})}{b_0} \geq \frac{H(1,\mathbf{y})}{b_1}} H(1, \mathbf{y})}{b_1} . \quad (3)$$

Then $BER(\mathbf{f}_H, \mathbf{b}) = \frac{\beta_H}{2} \leq \frac{1}{2}$, where equality holds iff f_H is constant. Hence, C is ε-fair iff $\beta_H > 2\varepsilon$.

Given the above theoretical preliminaries we now turn our attention to devising algorithms for solving Problem FRM.

4.2 Improving fairness

The algorithms that we propose are greedy. In each step, they will select a user who will be moved from one column of the contingency table to another (in the sense that his or her recommendation vector will be changed) so that the increase in BER will be maximal.

Formally, a *move* is a triplet of the form $(u, \mathbf{y}, \mathbf{z})$ where $u \in U$ is a user, $\mathbf{y} = C(u, \cdot)$ is the recommendation vector assigned by C to u, and $\mathbf{z} \in \{0,1\}^I \setminus \{\mathbf{y}\}$ is another recommendation vector. Each move represents a single action on the matrix C: applying the move $(u, \mathbf{y}, \mathbf{z})$ on C results with another matrix C' in which all rows are the same as in C, but the row corresponding to u is changed from $\mathbf{y}$ to $\mathbf{z}$. A move is called *fairness-improving* if the BER of the MBP that corresponds to C' is greater than the BER of the MBP corresponding to C.

Given a sequence of fairness-improving moves, where each user $u \in U$ appears in the sequence at most once, the matrix C' that the sequence induces is the matrix that is obtained from C after applying on it all moves in the sequence. Namely, $C'(u, \cdot) = \mathbf{z}$ for any move $(u, \mathbf{y}, \mathbf{z})$ in the sequence, while for all users who are not included in the sequence $C'(u, \cdot) = C(u, \cdot)$.

A solution to problem FRM is a sequence of fairness-improving moves for which the induced matrix C' is ε-fair.

We proceed to analyze the effect of single moves on the BER of the corresponding MBP, namely, on the fairness threshold of the recommendation matrix. Let $Y_0 := f_H^{-1}(0)$ be the subset of all recommendation vectors $\mathbf{y} \in Y$ (where Y is as in Definition 7) that f_H maps to 0, and $Y_1 := f_H^{-1}(1)$ be the complement subset. In the example in Table 1, there are $M = |I| = 2$ items and hence the recommendation matrix C has (at most) 4 types of rows, taken from $\{0,1\}^I$. In that example, $Y_0 = \{(0,0),(1,1)\}$ (namely, those are the recommendation vectors for which the MBP predicts “man”) and $Y_1 = \{(0,1),(1,0)\}$. So, we can read from H that the number of men for whom the vector $(0,1)$ was recommended is 5, while the number of women for whom that vector was recommended is 8.

H	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$\mathbf{b} = 0$	6	5	4	8
$\mathbf{b} = 1$	3	8	5	8

Table 1: A contingency table H

Let us denote, for $i \in \{0,1\}$, $x_i := \sum_{\mathbf{y} \in Y_0} H(i, \mathbf{y})$ and $y_i := \sum_{\mathbf{y} \in Y_1} H(i, \mathbf{y})$. That is, x_i is the number of users of gender i for whom the MBP predicts ”man”, while y_i is the number of users of gender i for whom the MBP predicts ”woman”. Then, by Definition 4,

$$BER := BER(\mathbf{f}_H, \mathbf{b}) = \frac{1}{2} \cdot \left[\frac{y_0}{b_0} + \frac{x_1}{b_1} \right] . \tag{4}$$

Assume that $BER < \varepsilon$ and that we aim to increase it by at least $d := \varepsilon - BER$ so that it becomes $\geq \varepsilon$. One way of doing that is to select a man for whom the recommendation vector was from Y_0 and change the latter to a vector of recommendations from Y_1; namely, to take a man for whom the MBP predicted correctly his gender, and change his recommendation vector to another vector on which the MBP predicts ”woman”, in order to increase the prediction error (and thus increase fairness towards our target threshold ε). The other option is to select a woman for whom the recommendation vector was from Y_1 and change the latter to a vector of recommendations from Y_0.

Next, we analyze the effects of moving a man from some $\mathbf{y}_0 \in Y_0$ to some $\mathbf{y}_1 \in Y_1$. To this end we split the columns of the contingency table H that correspond to vectors in Y_0 into two subgroups; we perform a similar split for the Y_1-columns.

Definition 9 A vector $\mathbf{y} \in Y_0$ is called *stable under removal of a man* if $\frac{H(0,\mathbf{y})-1}{b_0} \geq \frac{H(1,\mathbf{y})}{b_1}$ and unstable otherwise. A vector $\mathbf{y} \in Y_1$ is called *stable under addition of a man* if $\frac{H(0,\mathbf{y})+1}{b_0} < \frac{H(1,\mathbf{y})}{b_1}$ and unstable otherwise.

We proceed the explain the meaning of stability as defined above. If $\mathbf{y} \in Y_0$ it means that $f_H(\mathbf{y}) = 0$ and that happens iff $\frac{H(0,\mathbf{y})}{b_0} \geq \frac{H(1,\mathbf{y})}{b_1}$. The vector $\mathbf{y}$ is called stable under removal of a man if after removing from it one man, an action that will update the entry $H(0, \mathbf{y})$ to $H(0, \mathbf{y}) - 1$, it holds that $\frac{H(0,\mathbf{y})-1}{b_0} \geq \frac{H(1,\mathbf{y})}{b_1}$. In such a case, $f_H(\mathbf{y})$ still equals zero also after the move, hence the term “stable”. However, if $\frac{H(0,\mathbf{y})-1}{b_0} < \frac{H(1,\mathbf{y})}{b_1}$, then the value of $f_H(\mathbf{y})$ changes to 1 in wake of that move; namely, the value of f_H on $\mathbf{y}$ flips, hence the term “unstable” for such columns.

The next lemma spells out the effects of moves of a single man.

Lemma 1 *(1) A move of a man from a stable $\mathbf{y}_0 \in Y_0$ (under removal of a man) to a stable $\mathbf{y}_1 \in Y_1$ (under addition of a man) increases BER by $\frac{1}{2b_0}$.*

(2) A move of a man from a stable $\mathbf{y}_0 \in Y_0$ to an unstable $\mathbf{y}_1 \in Y_1$ increases BER by $\frac{1}{2} \cdot \left(\frac{H(1,\mathbf{y}_1)}{b_1} - \frac{H(0,\mathbf{y}_1)}{b_0} \right) \in (0, \frac{1}{2b_0}]$.

(3) A move of a man from an unstable $\mathbf{y}_0 \in Y_0$ to a stable $\mathbf{y}_1 \in Y_1$ increases BER by $\frac{1}{2} \cdot \left(\frac{H(0,\mathbf{y}_0)}{b_0} - \frac{H(1,\mathbf{y}_0)}{b_1} \right) \in [0, \frac{1}{2b_0})$.

(4) A move of a man from an unstable $\mathbf{y}_0 \in Y_0$ to an unstable $\mathbf{y}_1 \in Y_1$ adds to BER $\frac{1}{2} \cdot \left(\frac{H(0,\mathbf{y}_0)-1}{b_0} - \frac{H(1,\mathbf{y}_0)}{b_1}\right) - \frac{1}{2} \cdot \left(\frac{H(0,\mathbf{y}_1)}{b_0} - \frac{H(1,\mathbf{y}_1)}{b_1}\right) = \frac{\eta}{2b_0}$, where $-1 < \eta < 1$.

The next definition and lemma are the equivalent of Definition 9 and Lemma 1 for movings of women.

Definition 10 A vector $\mathbf{y} \in Y_1$ is called *stable under removal of a woman* if $\frac{H(0,\mathbf{y})}{b_0} < \frac{H(1,\mathbf{y})-1}{b_1}$ and unstable otherwise. A vector $\mathbf{y} \in Y_0$ is called *stable under addition of a woman* if $\frac{H(0,\mathbf{y})}{b_0} \geq \frac{H(1,\mathbf{y})+1}{b_1}$ and unstable otherwise.

Lemma 2 *(1) A move of a woman from a stable $\mathbf{y}_1 \in Y_1$ (under removal of a woman) to a stable $\mathbf{y}_0 \in Y_0$ (under addition of a woman) increases BER by $\frac{1}{2b_1}$.*

(2) A move of a woman from a stable $\mathbf{y}_1 \in Y_1$ to an unstable $\mathbf{y}_0 \in Y_0$ increases BER by $\frac{1}{2} \cdot \left(\frac{H(0,\mathbf{y}_0)}{b_0} - \frac{H(1,\mathbf{y}_0)}{b_1}\right) \in [0, \frac{1}{2b_1})$.

(3) A move of a woman from an unstable $\mathbf{y}_1 \in Y_1$ to a stable $\mathbf{y}_0 \in Y_0$ increases BER by $\frac{1}{2} \cdot \left(\frac{H(1,\mathbf{y}_1)}{b_1} - \frac{H(0,\mathbf{y}_1)}{b_0}\right) \in (0, \frac{1}{2b_1}]$.

(4) A move of a woman from an unstable $\mathbf{y}_1 \in Y_1$ to an unstable $\mathbf{y}_0 \in Y_0$ adds to BER $\frac{\eta}{2b_1}$, where $-1 < \eta < 1$.

Lemmas 1 and 2 imply the following summary of the effect of single moves on the BER:

Theorem 3 *For each $\mathbf{y} \in Y$ define $s(\mathbf{y}) := \frac{H(0,\mathbf{y})}{b_0} - \frac{H(1,\mathbf{y})}{b_1}$. Let $\mathbf{y}_0 \in Y_0$ and $\mathbf{y}_1 \in Y_1$. Then a move of a man from $\mathbf{y}_0$ to $\mathbf{y}_1$ increases BER by*

$$\Delta_m := \frac{1}{2} \cdot \left\{\min(s(\mathbf{y}_0), \frac{1}{b_0}) - \max(s(\mathbf{y}_1), -\frac{1}{b_0}) - \frac{1}{b_0}\right\},$$

while a move of a woman from $\mathbf{y}_1$ to $\mathbf{y}_0$ increases BER by

$$\Delta_w := \frac{1}{2} \cdot \left\{\min(s(\mathbf{y}_0), \frac{1}{b_1}) - \max(s(\mathbf{y}_1), -\frac{1}{b_1}) - \frac{1}{b_1}\right\}.$$

4.3 An optimal MBP-respecting solution with respect to μ_1

In order to achieve fairness to some level ε, it is needed to change the recommendation vectors to some users. Theorem 4 asserts that it is possible to achieve this goal by changing users' recommendation vectors only to other recommendation vectors that existed in the original recommendation matrix.

Theorem 4 *There exists an optimal solution matrix C' to Problem FRM with μ_1 where $Y' := \{C'(u,\cdot) : u \in U\} \subseteq Y := \{C(u,\cdot) : u \in U\}$.*

All the algorithms that we present produce moves and solutions that do not alter the MBP of the recommendation matrix. We refer to such moves and solutions as MBP-respecting, as defined next.

Definition 11 Assume that $\mathbf{y}, \mathbf{z} \in Y$ and that $u \in U$ is a user for whom $C(u,\cdot) = \mathbf{y}$. Let C' be the recommendation matrix that is obtained from C by the move $(u, \mathbf{y}, \mathbf{z})$, and let $Y' = \{C'(u,\cdot) : u \in U\}$. Finally, let $f_H : Y \to \{0,1\}$ and $f'_H : Y' \to \{0,1\}$ be the MBPs corresponding to C and C', respectively. Then the move is called *MBP-respecting* if f_H agrees with f'_H on Y'. A solution is called *MBP-respecting* if it consists only of MBP-respecting moves.

Two comments are in order before we proceed. First, it should be noted that either $Y' = Y$, or $Y' = Y \setminus \{\mathbf{y}\}$; the latter equality holds iff u was the only user for whom $C(u,\cdot) = \mathbf{y}$. A move is MBP-respecting if the MBP remains the same over the a-posteriori domain of definition, Y'. Second, in typical cases, where the exact same recommendation vector is not offered to more than one user, the subset of MBP-respecting solutions constitutes a sufficiently large playground (see Section 4.5).

Lemmas 1 and 2 imply that a fairness-improving and MBP-respecting move increases BER by either $1/2b_0$ (if u is a man) or $1/2b_1$ (if u is a woman). Assume, without loss of generality, that $b_0 \geq b_1$. In such a setting, if we wish to increase the BER to beyond a given threshold $\varepsilon < 1/2$, while affecting the least number of users, it is clear that it is needed to move first only women, and only after exhausting all possible women moves, if still necessary, start moving men.

Lemma 3 *The maximal number of women that can be moved without changing the MBP is*

$$\min\left\{\sum_{\mathbf{y}\in Y_0} \lfloor b_1 s(\mathbf{y})\rfloor, \sum_{\mathbf{y}\in Y_1} \lceil -b_1 s(\mathbf{y}) - 1\rceil\right\}. \tag{5}$$

The maximal number of men that can be moved without changing the MBP is

$$\min\left\{\sum_{\mathbf{y}\in Y_0} \lfloor b_0 s(\mathbf{y})\rfloor, \sum_{\mathbf{y}\in Y_1} \lceil -b_0 s(\mathbf{y}) - 1\rceil\right\}. \tag{6}$$

We are now ready to present Algorithm FaiRecSys (Algorithm 1) for solving Problem FRM with respect to μ_1. Let $C \in \{0,1\}^{U\times I}$ be a recommendation matrix, $\mathbf{b} \in \{0,1\}^U$ be a binary sensitive attribute vector, and $\varepsilon < \frac{1}{2}$ be a required fairness level. Algorithm FaiRecSys computes a recommendation matrix C' that is ε-fair, or a matrix C' that is δ-fair for a value of δ as high as possible to achieve by means of MBP-respecting moves only.

For the sake of simplicity we assume that $b_0 \geq b_1$. First, the algorithm calls the procedure MoveGender with the input parameters C, $\mathbf{b}$, and ε, and the gender indicator 1, since under the assumption $b_0 \geq b_1$ one starts with moving women (users of gender $i = 1$). That procedure performs the exact number of gender i moves which are MBP-respecting (i.e., from a stable column to another stable column) towards getting a BER of at least ε. It returns the modified C and the resulting BER value, δ. If the BER target is met, the algorithm stops. Otherwise, it performs a similar procedure with the gender indicator 0. At the end it returns the resulting recommendation matrix C and the achieved BER δ. If $\delta \geq \varepsilon$, then the output C is an optimal solution to the problem. Otherwise, C is the matrix that induces the same MBP as the original one, while maximizing the fairness level to as close as possible to ε.

Procedure MoveGender starts with initial computations (Steps 1-4). Then, it computes the required increase in BER into d (Step 5). If the current BER is already greater than or equal to ε, the procedure returns C and δ (Steps 5-7). It then computes the *capacities* of all vectors $\mathbf{y}$; specifically, in case $i = 1$ it computes for each $\mathbf{y} \in Y_0$ the maximal number $p(\mathbf{y})$ of women that this column can intake (Step 8), and for each $\mathbf{y} \in Y_1$ the maximal number $q(\mathbf{y})$ of women it can lose, without affecting the MBP. (When $i = 0$ then p and q denote the maximal number of men that $\mathbf{y}$ could lose or intake, respectively, while leaving the MBP unchanged.) Next, it computes from those capacities the overall number t of possible MBP-respecting moves of users of gender i, as implied by Lemma 3 (Step 10). It then computes the number ℓ of actual moves, as the minimum between t and the number of moves that are needed in order to increase BER by d (Step 11). It then moves ℓ users of gender i from vectors in Y_i to vectors in Y_{1-i} in compliance with the computed capacities; namely, it moves ℓ users of gender i so that each vector $\mathbf{y} \in Y_i$ loses no more than the number of users that it can lose without altering the MBP, and each vector $\mathbf{y} \in Y_{1-i}$ intakes no more than the number of users that it can take in without altering the MBP (Step 12). Finally, δ is updated to reflect the increase in BER that was caused by those moves and the algorithm returns C and δ (Steps 13-14).

Theorem 5 *Let $\mathcal{M}(C)$ denote the set of all matrices that can be obtained from C by means of MBP-respecting moves only. Let $\mathcal{M}(C, \varepsilon)$ denote the subset of $\mathcal{M}(C)$ including all matrices that are ε-fair. Let C' be the matrix that Algorithm* FaiRecSys *outputs and δ be the BER of the corresponding MBP. Then if $\delta \geq \varepsilon$, C' is a matrix in $\mathcal{M}(C, \varepsilon)$ which minimizes dist(C, C'), Eq. (2), with $\mu = \mu_1$. If $\delta < \varepsilon$, then $\mathcal{M}(C, \varepsilon) = \emptyset$ and C' is a matrix in $\mathcal{M}(C)$ for which the BER is maximal.*

Algorithm 1 FaiRecSys: Solving Problem FRM with μ_1

```
Input: $C$, $\mathbf{b}$, $\varepsilon$. (Assumption: $b_0 \geq b_1$)
Output: A solution to Problem FRM with $\mu_1$.
1: MoveGender($C, \mathbf{b}, \varepsilon, 1, \delta$).
2: if $\delta \geq \varepsilon$ then
3:     stop
4: end if
5: MoveGender($C, \mathbf{b}, \varepsilon, 0, \delta$).
6: Output $C$, $\delta$
```

Algorithm 2 Procedure MoveGender

```
Input: $C$ (the initial matrix), $\mathbf{b}$, $\varepsilon$ (the target BER), $i$ (the gender to move)
Output: $C$ (the resulting matrix), $\delta$ (the a-posteriori BER)
1: Compute the set of recommendation vectors $Y$ and the contingency table $H$.
2: Compute $s(\mathbf{y}) = \frac{H(0,\mathbf{y})}{b_0} - \frac{H(1,\mathbf{y})}{b_1}$ for all $\mathbf{y} \in Y$.
3: Set $Y_0 := \{\mathbf{y} : s(\mathbf{y}) \geq 0\}$ and $Y_1 := \{\mathbf{y} : s(\mathbf{y}) < 0\}$.
4: Compute $\delta := \frac{1}{2} \cdot \left( \frac{\sum_{\mathbf{y} \in Y_0} H(1,\mathbf{y})}{b_1} + \frac{\sum_{\mathbf{y} \in Y_1} H(0,\mathbf{y})}{b_0} \right)$.
5: if $d := \varepsilon - \delta \leq 0$ then
6:     return $C$ and $\delta$
7: end if
8: For each $\mathbf{y} \in Y_0$ compute $p(\mathbf{y}) := \lfloor b_i s(\mathbf{y}) \rfloor$.
9: For each $\mathbf{y} \in Y_1$ compute $q(\mathbf{y}) := \lceil -b_i s(\mathbf{y}) - 1 \rceil$.
10: $t := \min \left\{ \sum_{\mathbf{y} \in Y_0} p(\mathbf{y}), \sum_{\mathbf{y} \in Y_1} q(\mathbf{y}) \right\}$.
11: $\ell := \min\{\lceil 2 b_i d \rceil, t\}$.
12: Move $\ell$ users of gender $i$ from $Y_i$ to $Y_{1-i}$
13: $\delta = \delta + \frac{\ell}{2 b_i}$.
14: return $C$ and $\delta$
```

We finally discuss the complexity of Algorithm FaiRecSys. That algorithm only invokes procedure MoveGender, either once or twice. The complexity of procedure MoveGender is dominated by the complexity of the first step in it, which is $O(NM)$ (recall that N is the number of users while M is the number of items). The complexity of all subsequent steps in MoveGender is $O(N)$, since the number of distinct vectors in Y is bounded by N.

4.4 Extending the algorithm for a general metric

Algorithm FaiRecSys was guided by the binary metric μ_1, where $\mu_1(\mathbf{y}, \mathbf{z}) = 1$ whenever $\mathbf{y} \neq \mathbf{z}$. Here, we discuss its extension to more sensitive metrics, like μ_2. Such metrics require a more careful execution of Step 12 in procedure MoveGender, towards minimizing the induced distance between the a-priori and a-posteriori recommendation matrices. We proceed to discuss the implementation of Step 12 for such metrics. The rest of the algorithm remains the same.

First, for each $\mathbf{y} \in Y_0$ and $\mathbf{z} \in Y_1$ we compute $\mu_2(\mathbf{y}, \mathbf{z})$. Then, Step 12 raises the need to solve the following optimization problem. We focus on the case $i = 1$, where it is needed to move women. In that case, each $\mathbf{y} \in Y_1$ has a number $q(\mathbf{y})$ (Step 9) that indicates the number of women that were assigned $\mathbf{y}$ and for whom we need to change the recommendation vector to some vector in Y_0. For each $\mathbf{z} \in Y_0$ we have a number $p(\mathbf{z})$ (Step 8) that indicates the number of times that it can be used in such replacements. The goal is to find ℓ pairs, $L := \{(\mathbf{y}_j \in Y_1, \mathbf{z}_j \in Y_0) : 1 \leq j \leq \ell\}$, where ℓ is as computed in Step 11, so that: (a) no $\mathbf{y} \in Y_1$ appears in L more than $q(\mathbf{y})$ times; (b) no $\mathbf{z} \in Y_0$ appears in L more than $p(\mathbf{z})$ times; and (c) $\sum_{j=1}^{\ell} \mu_2(\mathbf{y}_j, \mathbf{z}_j)$ is minimal. This min-cost ℓ-flow problem can be solved optimally as follows.

Let V_0 be a set that holds $p(\mathbf{y})$ copies of each vector $\mathbf{y} \in Y_0$ and V_1 be a set that holds $q(\mathbf{z})$ copies for each vector $\mathbf{z} \in Y_1$. (Note that $\min\{|V_0|, |V_1|\} = t$ for t as defined in Step 10.) Consider the complete directed bipartite graph over V_0 and V_1 where for each node $\mathbf{y} \in V_0$ and $\mathbf{z} \in V_1$, the directed edge $(\mathbf{y}, \mathbf{z})$ has cost $\mu_2(\mathbf{y}, \mathbf{z})$. Let $\mathbf{x}_s$ be an external source node, where for each $\mathbf{y} \in V_0$ there is a directed edge $(\mathbf{x}_s, \mathbf{y})$ of cost 0. Similarly, let $\mathbf{x}_t$ be an external sink node with 0-cost directed edges $(\mathbf{z}, \mathbf{x}_t)$ for all $\mathbf{z} \in V_1$. Then we may invoke an algorithm for finding an *integral* ℓ-flow in that graph with minimal cost. Such an ℓ-flow dictates ℓ moves of women to be implemented in Step 12 in procedure MoveGender.

4.5 The case of unique recommendations

One may view the recommendation matrix C as a mapping from U to the subset of $\{0,1\}^I$ consisting of all k-weighted vectors. Since the size of the target set is very large, $\binom{M}{k}$, then usually C is one-to-one. Namely, no two users are offered the same recommendation vector. In such cases, $|Y| = N$, and the contingency table H consists of b_0 columns that equal $(1,0)^T$ (corresponding to recommendation vectors that were offered to a single man) and b_1 columns that equal $(0,1)^T$. The MBP in this case predicts $\mathbf{b}$ perfectly; Y_i is then the collection of b_i vectors offered to users of gender $i \in \{0,1\}$.

Lemma 4 *Assume that $b_0 \geq b_1$ and let $r := b_0 \bmod b_1$. Then by considering only MBP-improving and respecting moves, it is possible to reach a BER of $\frac{1}{2} \cdot \frac{b_0 - r}{b_0}$. Furthermore, the latter value is always $> 1/4$.*

Recall that the maximal possible BER of the MBP is 1/2, which is the BER of a naïve constant predictor (Theorem 2). Lemma 4 shows that by restricting our attention to MBP-respecting moves only, we are still able to meet the ε-fairness for $\varepsilon \leq \frac{1}{2} \cdot \frac{b_0 - r}{b_0}$ (assuming that $b_0 \geq b_1$). In some cases, the latter upper bound is close to 1/2 (for example, if b_0 and b_1 are large and $b_0 \approx b_1$, or if $b_0 \gg b_1$). But, in any case, it is always possible to achieve ε-fairness for any $\varepsilon \leq 1/4$.

5 Experiments

This section reports the experimental evaluation of the effects of Algorithm FaiRecSys on the quality/utility of recommendations, as well as of the achieved bias reduction.

5.1 Experimental setup

We describe herein our experimental setup: datasets, preprocessing methodology, fairness characteristics of the datasets, and implementation details.

For our experiments, we used two real-world datasets, **MovieLens** and **Reddit**: **MovieLens** from the Grouplens research team[3] [15] is a well known dataset, which typically used in recommender system literature. It consists of 1M ratings in the range $[1, 5]$. Some demographics such as gender, age and occupation are also provided for each user. We adopted the convention in previous studies that ratings above 3 are considered as positive feedbacks [24]. After removing the ratings below 3, $|U| = 6040$ users (4331 male and 1709 female) and $|I| = 3534$ items remained in the dataset. The **Reddit**[4] dataset was described in Section 1.1. We used **Reddit**'s open API[5] to retrieve all 934,015,622 comments from years 2013 and 2014 submitted at 166,742 subreddits from 8,668,780 users. We selected $|U| = 32,148$ users that reported their gender in some subreddits that support gender self-reporting. They correspond to 20,371 males and 11,777 females who submitted 30,150,270 comments in $|I| = 24,112$ subreddits.

Since gender information is available in both datasets, we used the gender as the sensitive attribute. Nevertheless, our approach can be applied to any binary sensitive attribute. For both datasets we generated a binary matrix $R = (r_{u,i} : u \in U, i \in I)$ with implicit feedback entries, where $r_{u,i} = 1$ if user u posted a comment in subreddit i (in **Reddit**) or if user u rated the movie i above 3 (in **MovieLens**), and 0 otherwise.

Let $I(u) := \{i \in I : R(u,i) = 1\}$ be the set of items for which user u showed positive preference. The

[3] http://grouplens.org/datasets/movielens/
[4] http://www.reddit.com
[5] https://www.reddit.com/dev/api/

training set $I_{\text{train}}(u)$ for user u was set to be a random subset of $I(u)$ of size $|I(u)|/2$. The remaining preferences, $I_{\text{test}}(u) := I(u) \setminus I_{\text{train}}(u)$, constituted the test set. The training set is used to fit a *weighted regularized matrix factorization* (WRMF) recommendation model [16], which is appropriate for implicit feedbacks. The model parameters were selected using 10-fold cross-validation. After training, we generated for each user u prediction scores for all items outside $I_{\text{train}}(u)$. Finally, the k items with the highest prediction scores were recommended to user u, and $C = (c_{u,i} : u \in U, i \in I)$ was built as in Definitions 1-2 for different values of k.

Next, we discuss the manner in which we estimated the bias in the matrices C and C'. As discussed in Section 3, $BER(\mathbf{f}(C), \mathbf{b})$ measures the level of bias. In view of Theorem 1, the MBP, f_H, is the most accurate predictor as no predictor attains a BER smaller than $BER(\mathbf{f}_H, \mathbf{b})$. Namely, by measuring $BER(\mathbf{f}_H, \mathbf{b})$ we get an upper bound for the bias in C. Since the MBP is a theoretical predictor, we also considered a realistic predictor, Random Forest Predictor [1](RFP), to predict $\mathbf{b}$ from C. A Random Forest Predictor $g : \{0,1\}^I \to [0,1]$ is an ensemble of decisions trees. For a given test point $C(u,\cdot)$, it outputs the average predicted probabilities $P[b = 1|C(u,\cdot)]$ of all the trees in the ensemble. C and $\mathbf{b}$ are split into 60% for $(C_{train}, \mathbf{b}_{train})$, 10% for $(C_{val}, \mathbf{b}_{val})$ and 30% for $(C_{test}, \mathbf{b}_{test})$. A model was trained using $(C_{train}, \mathbf{b}_{train})$, and then applied on C_{val} in order to produce the probabilities of $\mathbf{b}_{val}$ being 0 or 1. Different thresholds were applied to the probabilities to produce $BER(\mathbf{g}(C)_{val}, \mathbf{b}_{val})$. The threshold that produces the minimum BER was selected, binary predictions were produced for C_{test}, and finally $BER(\mathbf{g}(C_{test}), \mathbf{b}_{test})$ is reported.

We used the implementation of WRMF provided by *mrec* recommender system library[6] [23]. For the solution of the min-cost flow problems (see Section 4.4), we used Google Optimization Tools library[7] [10,8,9]. Scikit-learn[8] [27] machine learning library was used for the Random Forest Predictor. Our proposed algorithm and evaluation metrics were implemented in Python[9]. Finally, in the experiments that are based on the distance metric μ_1, Algorithm FaiRecSys chose vectors for replacement in a random manner. We repeated those experiments 10 times and reported averaged results.

We recall that if the inputs to Algorithm FaiRecSys are C and ε, then the outputs are a modified recommendation matrix C' and a parameter δ that equals $BER(\mathbf{f}_H(C'), \mathbf{b})$. Ideally, $\delta \geq \varepsilon$. However, it is possible that $\delta < \varepsilon$. The latter case occurs when it is not possible to increase the MBP-BER as high as ε, and then δ is the maximum value of MBP-BER which can be achieved by our approach. In all of our experiments, we had $\delta \geq \varepsilon$ (namely, we were able to reach the desired level of fairness). Therefore, we use all over ε to denote the MBP-BER of the output matrix C'.

	MovieLens			**Reddit**		
	$k = 10$	$k = 20$	$k = 50$	$k = 10$	$k = 20$	$k = 50$
$BER(\mathbf{f}_H(C), \mathbf{b})$	0	0	0	0.0001	0	0
$BER(\mathbf{g}(C_{test}), \mathbf{b}_{test})$	0.32	0.31	0.29	0.17	0.12	0.09

Table 2: BER values for the original datasets.

5.2 Preliminary results

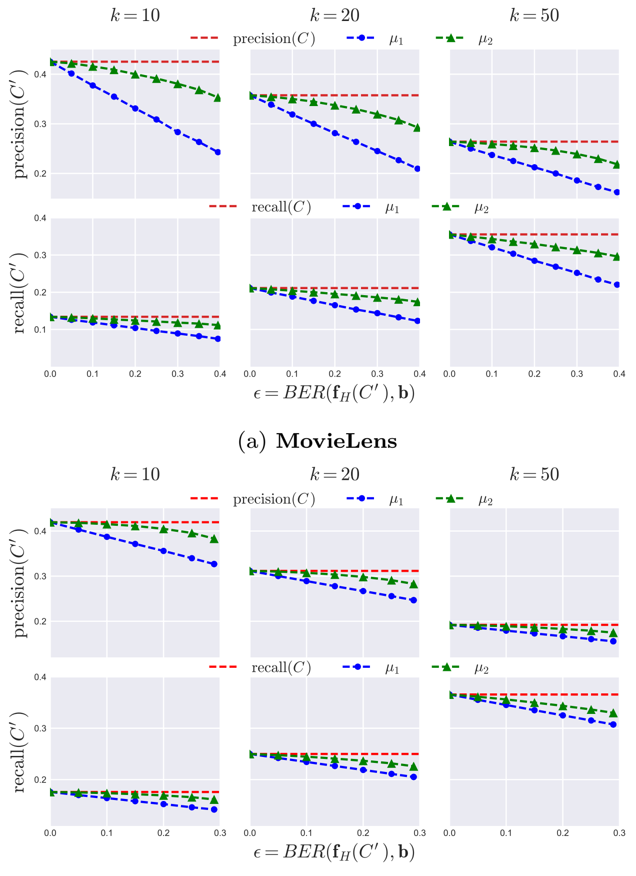
Table 2 presents the BER values that were computed from the original recommendation matrices of both datasets, for different values of k and both MBP and RFP. For all pairs $(dataset, k)$, each user has a unique recommendation vectors, except (**Reddit**, $k = 10$) where more than 99% of users have a unique recommendation vector. Thus, as discussed in Section 4.5, when each user has a unique recommendation vector, the BER of the MBP(MBP-BER) is 0. In agreement with Theorem 1, the BER values of the MBP are always smaller than those of the RFP. We also note that the **Reddit** dataset has smaller values of BER for RFP(RFP-BER) compared to **MovieLens**, suggesting that **Reddit** is more biased than **MovieLens**.

The maximum value of MBP-BER that we can get in Algorithm FaiRecSys's output recommendation matrix is given in Lemma 4. For **MovieLens**, as $b_0 = 4331$ and $b_1 = 1709$, we have $q = 2$ and, hence, $r = 4331 - 2 \cdot 1709 = 913$. By Lemma 4, the maximum possible value of MBP-BER is therefore $(0.5 \cdot \frac{4331-913}{4331}) = 0.3946$. For **Reddit**, the maximum value of MBP-BER is 0.2891, except for the case where $k = 10$ in which not all recommendation vectors are unique; in that case, the maximum value of MBP-BER is 0.2892.

5.3 Measuring utility

The first objective of our experimental evaluation is to show the effects of our transformations on the quality/utility of the aposteriori recommendations, C', compared to the original ones, C. In order to compare the quality of original recommendation matrix C with the transformed (and less biased) matrix C', we computed the utility measures *precision* and *recall* from C and

[6] http://mendeley.github.io/mrec/
[7] https://developers.google.com/optimization
[8] http://scikit-learn.org/
[9] https://www.python.org



(a) **MovieLens**

(b) **Reddit**

Fig. 1: *precision*(C') and *recall*(C') for different values of ε and k for **MovieLens** and **Reddit** datasets.

from C', for varying values of k and ε. Letting $I_{\text{rec}}(u)$ denote the subset of items that are recommended to user u based on the training set, then *precision* is defined as $\frac{1}{|U|}\sum_{u\in U}\frac{|I_{\text{rec}}(u)\cap I_{test}(u)|}{|I_{\text{rec}}(u)|}$, while *recall* is defined as $\frac{1}{|U|}\sum_{u\in U}\frac{|I_{\text{rec}}(u)\cap I_{test}(u)|}{|I_{test}(u)|}$.

Figure 1 shows the precision and recall values, as obtained from C', for different values of ε and k, using both μ_1, μ_2. The red dashed lines show the upper bound for precision and recall, as calculated from the original recommendation matrix C. As expected, the values of precision and recall decrease for increasing values of ε, since larger values of ε imply higher fairness and, consequently, higher prices of fairness. Moreover, generating the recommendation matrices C' with μ_2 leads to higher precision and recall than with μ_1. This trend is observed in both datasets.

5.4 Measuring bias

Herein we evaluate the level of bias which is achieved by our algorithm for different values of ε and k, when the guiding metric is either μ_1 or μ_2. We computed $BER(\mathbf{f}_H(C'), \mathbf{b})$ (the a-posteriori MBP-BER) and compared it with $BER(\mathbf{f}_H(C), \mathbf{b})$ (the original MBP-BER). We repeat those evaluations with the BER achieved by the Random Forest Predictor (RFP). Figure 2 shows the correlation between RFP-BER, $BER(\mathbf{g}(C'_{test}), \mathbf{b}_{test})$), and MBP-BER, $BER(\mathbf{f}_H(C'), \mathbf{b})$. The RFP results are important in order to understand the effects of applying our fairness algorithm in a realistic scenario. While the use of MBP is important in developing a theoretical framework, it might be less relevant for evaluating bias in a realistic scenario. Therefore, the use of RFP, alongside the MBP, sheds light on the relation between theory and practice.

For the **Reddit** dataset, both BER values increase when the input parameter ε is increased. Also, it can be observed that using the metric μ_2 brings about lower $BER(\mathbf{g}(C'_{\text{test}}), \mathbf{b}_{test})$ values than the corresponding ones for μ_1. On the other hand, for the **MovieLens** dataset we observed that $BER(\mathbf{g}(C'_{\text{test}}), \mathbf{b}_{test}) = 0.5$ when $\mu = \mu_1$ and $BER(\mathbf{f}_H(C'), \mathbf{b}) > 0.15$. For μ_2, only when $BER(\mathbf{f}_H(C'), \mathbf{b}) > 0.3$ $BER(\mathbf{g}(C'_{\text{test}}), \mathbf{b}_{test})$ reaches the upper limit of 0.5. These findings show two things:

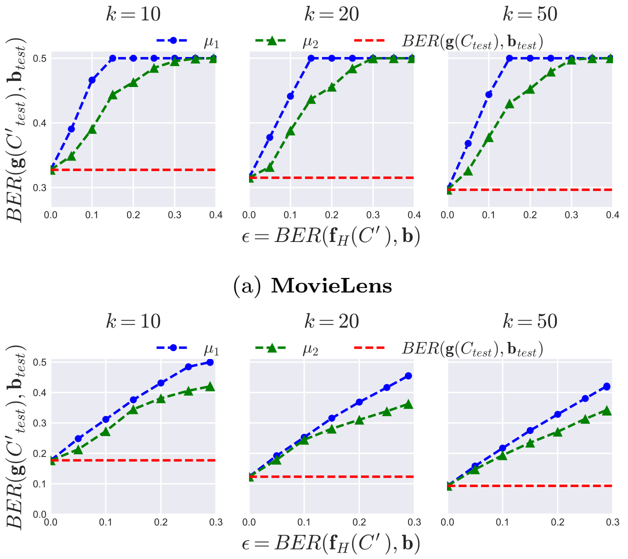
(1) There is a trade-off between the two metrics μ_1 and μ_2. While the loss in recommendation quality is higher when using μ_1 (as shown in Figure 1), using μ_1 yields less biased recommendations (Figure 2).

(2) The bias in **Reddit** is stronger than that in **MovieLens**, since both BER curves for **MovieLens** reach the upper limit of 0.5 after introducing less changes in the recommendation matrix, in comparison to **Reddit** (Figure 2).

As another way to measure the bias achieved by FaiRecSys, we also computed the *item-wise bias*, denoted diff$_i$., Eq. (7): diff$_i$ equals the difference between the fraction of men for whom item i was recommended and the corresponding fraction of women. Optimal fairness for item i occurs when diff$_i = 0$.

$$\text{diff}_i = \frac{|\{u \in U : C(u,i) = 1 \wedge b(u) = 0\}|}{|\{u \in U : b(u) = 0\}|} - \frac{|\{u \in U : C(u,i) = 1 \wedge b(u) = 1\}|}{|\{u \in U : b(u) = 1\}|} \quad (7)$$

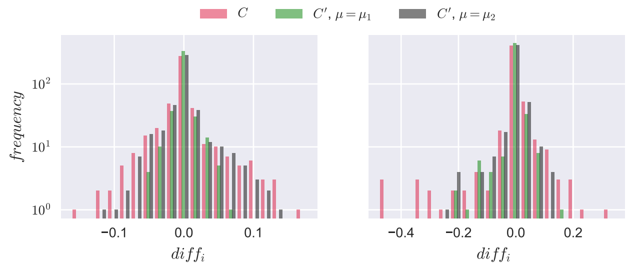
Figure 3 shows the distribution of the item-wise bias metric diff$_i$ for $k = 20$ and $\varepsilon = BER(\mathbf{f}_H(C'), \mathbf{b}) = 0.3$.



(a) **MovieLens**

(b) **Reddit**

Fig. 2: Effects of Algorithm FaiRecSys on the achieved level of bias, measured by MBP and RFP.



(a) **MovieLens**

(b) **Reddit**

Fig. 3: Effect of Algorithm FaiRecSys on item-wise bias metric diff_i

(Other settings of (k, ϵ) yield similar trends.) We compared the distributions of diff_i generated from the original matrix C, C' using $\mu = \mu_1$ and C' using $\mu = \mu_2$. We observed that the former distribution is more concentrated around 0, what reflect the smaller biases conveyed by $(C', \mu = \mu_1)$ and $(C', \mu = \mu_2)$ in comparison to C . For **MovieLens** dataset, we observe that diff_i values for $(C', \mu = \mu_1)$ are more concentrated around 0 than diff_i values for $(C', \mu = \mu_2)$; namely, in this dataset $(C', \mu = \mu_1)$ carries less item-wise bias. In case of the **Reddit** dataset, we observe smaller differences between μ_1 and μ_2. Moreover, when we compare Figure 3(a) and Figure 3(b), we also confirm that the **Reddit** dataset carries higher biases than the **MovieLens** dataset, since the distribution of diff_i is more concentrated around 0 for **MovieLens**.

6 Conclusions

The problem of algorithmic bias and the possibility that decisions informed by data mining algorithms may strengthen and perpetuate the bias existing in the training data, is nowadays recognized as one of the key problems for our community.

In this paper, motivated by empirical examples, we address the problem of algorithmic bias in recommender systems. We formulate a fairness constraint based on the notion of *predictability of sensitive features* (such as gender or ethnicity) and bias in the results of recommendations. We then propose FaiRecSys – an algorithm that mitigates algorithmic bias by post-processing the recommendation matrix with minimum impact on the utility of recommendations provided to the end-users.

In the future extended version of this work we are going to generalize the discussion from a binary sensitive attribute to sensitive attributes with larger finite domains (such as ethnicity or religion). Another extension, which is much more intricate, is to achieve simultaneous fairness with respect to several sensitive attributes (of any cardinalities). This extension is most essential, since users whose preferences are conveyed through the recommendation matrix may belong to possibly-discriminated groups based on more than one sensitive attribute. Clearly, achieving simultaneous fairness with respect to a given set of sensitive attributes may be impossible. Hence, an important first step in addressing this multi-dimensional challenge is to carefully formalize the inputs and desired outputs of the corresponding computational problem.

References

1. L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
2. R. Burke. Multisided fairness for recommendation. *CoRR*, abs/1707.00093, 2017.
3. T. Calders and S. Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.
4. A. Datta, M. C. Tschantz, and A. Datta. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112, 2015.
5. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012.
6. M. D. Ekstrand and M. S. Pera. The demographics of cool. In *Poster Proceedings at ACM RecSys. ACM, Como, Italy*, 2017.
7. M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

8. A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of algorithms*, 22(1):1–29, 1997.
9. A. V. Goldberg and M. Kharitonov. *On implementing scaling push-relabel algorithms for the minimum-cost flow problem*, volume 12. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1993.
10. A. V. Goldberg and R. E. Tarjan. Finding minimum-cost circulations by successive approximation. *Mathematics of Operations Research*, 15(3):430–466, 1990.
11. S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *KDD, 2016*.
12. S. Hajian and J. Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 25(7):1445–1459, 2013.
13. S. Hajian, J. Domingo-Ferrer, and O. Farràs. Generalization-based privacy preservation and discrimination prevention in data publishing and mining. *Data Mining and Knowledge Discovery*, 28(5-6):1158–1188, 2014.
14. S. Hajian, J. Domingo-Ferrer, A. Monreale, D. Pedreschi, and F. Giannotti. Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery*, 29(6):1733–1782, 2015.
15. F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.
16. Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee, 2008.
17. F. Kamiran and T. Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.
18. F. Kamiran, T. Calders, and M. Pechenizkiy. Discrimination aware decision tree learning. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 869–874. IEEE, 2010.
19. F. Kamiran, A. Karim, and X. Zhang. Decision theory for discrimination-aware classification. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 924–929. IEEE, 2012.
20. T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma. Enhancement of the neutrality in recommendation. In *The 2nd Workshop on Human Decision Making in Recommender Systems (Decisions)*, 2012.
21. T. Kamishima, S. Akaho, H. Asoh, and I. Sato. Model-based approaches for independence-enhanced recommendation. In *The IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 860–867, 2016.
22. Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.
23. M. Levy and K. Jack. Efficient top-n recommendation by linear regression. In *Proceedings of Large Scale Recommender System Workshop at ACM RecSys. ACM, Hong Kong, China*, 2013.
24. D. Lim, J. McAuley, and G. Lanckriet. Top-n recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 309–312. ACM, 2015.
25. B. T. Luong, S. Ruggieri, and F. Turini. 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*, pages 502–510. ACM, 2011.
26. K. Mancuhan and C. Clifton. Combating discrimination using bayesian networks. *Artificial intelligence and law*, 22(2):211–238, 2014.
27. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
28. D. Pedreschi, S. Ruggieri, and F. Turini. Measuring discrimination in socially-sensitive decision records. In *SDM*, pages 581–592. SIAM, 2009.
29. D. Pedreshi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568. ACM, 2008.
30. S. Ruggieri, S. Hajian, F. Kamiran, and X. Zhang. Anti-discrimination analysis using privacy attack strategies. In *Machine Learning and Knowledge Discovery in Databases*, pages 694–710. Springer, 2014.
31. S. Ruggieri, D. Pedreschi, and F. Turini. Data mining for discrimination discovery. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(2):9, 2010.
32. L. Sweeney. Discrimination in online ad delivery. *Queue*, 11(3):10, 2013.
33. R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 325–333, 2013.

A Proofs

A.1 Proof of Theorem 1

Fix $f \in \Pi$ and let Y_f be the subset of all vectors $\mathbf{y} \in \{0,1\}^I$ for which $f(\mathbf{y}) \neq f_H(\mathbf{y})$. Let us define $Y_0 = f_H^{-1}(0) \setminus Y_f$ and $Y_1 = f_H^{-1}(1) \setminus Y_f$. Therefore, Y_0 is the subset of all recommendation vectors $\mathbf{y} \in \{0,1\}^I$ that both f_H and f map to 0, Y_1 is the subset of all recommendation vectors $\mathbf{y} \in \{0,1\}^I$ that both f_H and f map to 1, and Y_f is the complementing subset, where f_H and f disagree.

Given the partition of columns into the above mentioned three subsets, $\{0,1\}^I = Y_0 \bigcup Y_1 \bigcup Y_f$, we denote the summation of cells of row number $i = 0, 1$ in the contingency table H in each of those subsets as follows:

$$r_i := \sum_{\mathbf{y} \in Y_0} H(i, \mathbf{y}) \,,$$

$$s_i := \sum_{\mathbf{y} \in Y_1} H(i, \mathbf{y}) \,,$$

$$t_i := \sum_{\mathbf{y} \in Y_f} H(i, \mathbf{y}) \,.$$

Recall that b_0 and b_1 denote the number of men and women, respectively, and therefore $r_0 + s_0 + t_0 = b_0$ and $r_1 + s_1 + t_1 = b_1$.

Let us assume first that Y_f includes just one vector $\mathbf{y}$. There are two possible cases regarding the value that f_H and f assign to $\mathbf{y}$: If $\frac{t_0}{b_0} \geq \frac{t_1}{b_1}$ then $f_H(\mathbf{y}) = 0$ and, hence, $f(\mathbf{y}) = 1$. Therefore,

$$2BER(\mathbf{f}_H, \mathbf{b}) = Pr[\mathbf{f}_H = 1 | \mathbf{b} = 0] + Pr[\mathbf{f}_H = 0 | \mathbf{b} = 1] = \frac{s_0}{b_0} + \frac{r_1 + t_1}{b_1} \,,$$

while

$$2BER(\mathbf{f}, \mathbf{b}) = Pr[\mathbf{f} = 1|\mathbf{b} = 0] + Pr[\mathbf{f} = 0|\mathbf{b} = 1] = \frac{s_0 + t_0}{b_0} + \frac{r_1}{b_1}.$$

Comparing the last two equalities we infer that $BER(\mathbf{f}, \mathbf{b}) \geq BER(\mathbf{f}_H, \mathbf{b})$ in this case. If, on the other hand, $\frac{t_0}{b_0} < \frac{t_1}{b_1}$ then $f_H(\mathbf{y}) = 1$ and, hence, $f(\mathbf{y}) = 0$. Therefore,

$$2BER(\mathbf{f}_H, \mathbf{b}) = Pr[\mathbf{f}_H = 1|\mathbf{b} = 0] + Pr[\mathbf{f}_H = 0|\mathbf{b} = 1] = \frac{s_0 + t_0}{b_0} + \frac{r_1}{b_1},$$

while

$$2BER(\mathbf{f}, \mathbf{b}) = Pr[\mathbf{f} = 1|\mathbf{b} = 0] + Pr[\mathbf{f} = 0|\mathbf{b} = 1] = \frac{s_0}{b_0} + \frac{r_1 + t_1}{b_1}.$$

Comparing the last two equalities we infer that $BER(\mathbf{f}, \mathbf{b}) \geq BER(\mathbf{f}_H, \mathbf{b})$ in this case as well.

This concludes the proof that $BER(\mathbf{f}_H, \mathbf{y}) \leq BER(\mathbf{f}, \mathbf{y})$ when $|Y_f| = 1$. The proof for $|Y_f| > 1$ goes along the same lines. □

A.2 Proof of Theorem 2

It is easy to see that the first addend on the right hand side of Eq. (3) equals $Pr[\mathbf{f}_H = 1|\mathbf{b} = 0]$, since the denominator in that fraction equals b_0 — the total number of men ($b(u) = 0$), while the numerator equals the total number of men for whom $f_H(u) = 1$. Similarly, the second addend on the right hand side of Eq. (3) equals $Pr[\mathbf{f}_H = 0|\mathbf{b} = 1]$. Hence, by Eq. (1), $BER(\mathbf{f}_H, \mathbf{b}) = \frac{\beta_H}{2}$.

In order to prove the upper bound of $\frac{1}{2}$, we introduce the following notations for $i = 0, 1$:

$$r_i := \sum_{\mathbf{y} \in f_H^{-1}(0)} H(i, \mathbf{y}), \quad s_i := \sum_{\mathbf{y} \in f_H^{-1}(1)} H(i, \mathbf{y}).$$

Namely, r_0 is the total number of men that f_H predicted to be men while r_1 is the total number of women that f_H predicted to be men. Similarly, s_0 is the total number of men that f_H predicted to be women while s_1 is the total number of women that f_H predicted to be women. With these notations, we infer from our discussion above that

$$BER(\mathbf{f}_H, \mathbf{b}) = \frac{1}{2} \cdot \left[\frac{s_0}{r_0 + s_0} + \frac{r_1}{r_1 + s_1} \right].$$

As, by the definition of f_H, $\frac{s_0}{b_0} \leq \frac{s_1}{b_1}$, where $b_0 = r_0 + s_0$ and $b_1 = r_1 + s_1$ are the total numbers of men and women, respectively, we infer that

$$BER(\mathbf{f}_H, \mathbf{b}) = \frac{1}{2} \cdot \left[\frac{s_0}{b_0} + \frac{r_1}{b_1} \right] \leq \frac{1}{2} \cdot \left[\frac{s_1}{b_1} + \frac{r_1}{b_1} \right] = \frac{1}{2} \cdot \frac{b_1}{b_1} = \frac{1}{2}. \tag{8}$$

It is easy to see that the inequality in Eq. (8) holds in the strict sense unless $f_H \equiv 0$ (in which case $s_0 = s_1 = 0$) or $f_H \equiv 1$ (in which case $r_0 = r_1 = 0$).

Finally, the last claim that C is ε-fair iff $\beta_H > 2\varepsilon$ follows directly from Theorem 1 and the equality $BER(\mathbf{f}_H, \mathbf{b}) = \frac{\beta_H}{2}$ which we just proved. That concludes the proof. □

A.3 Proof of Lemma 1

The first case is trivial since here f_H does not change, so BER will increase by $\frac{1}{2b_0}$ since the move will make f_H wrong for the man that was moved (whereas before f_H was right for that man).

To prove the second claim we look at the contingency table before and after the move, H and H' respectively, in a summarized manner as shown in Tables 3 and 4.

The notation # is used in all tables in this proof which show the contingency table in a summarized manner in order to mark the entries that contribute to the BER. So

$$2BER_H = \frac{y_0 + v_0}{b_0} + \frac{u_1 + x_1}{b_1}$$

while

$$2BER_{H'} = \frac{v_0}{b_0} + \frac{u_1 + x_1 + y_1}{b_1}.$$

We infer that

$$BER_{H'} - BER_H = \frac{1}{2} \cdot \left(\frac{y_1}{b_1} - \frac{y_0}{b_0} \right)$$

as claimed. That difference is indeed positive since, from the fact that $\mathbf{y}_1 \in Y_1$ (prior to the move) we know that $f_H(\mathbf{y}_1) = 1$, namely, that $\frac{y_1}{b_1} > \frac{y_0}{b_0}$. On the other hand, since $\mathbf{y}_1$ is unstable under addition of a man, we know that after the move $\frac{y_0+1}{b_0} \geq \frac{y_1}{b_1}$. But this implies that $\frac{1}{2} \cdot \left(\frac{y_1}{b_1} - \frac{y_0}{b_0} \right) \leq \frac{1}{2b_0}$, in accord with our claim.

The proof of the third case is similar. Here the contingency table H' is as given in Table 5. So now

$$2BER_{H'} = \frac{x_0 + y_0 + v_0}{b_0} + \frac{u_1}{b_1},$$

and, therefore,

$$BER_{H'} - BER_H = \frac{1}{2} \cdot \left(\frac{x_0}{b_0} - \frac{x_1}{b_1} \right) \geq 0.$$

That difference is strictly smaller than $\frac{1}{2b_0}$ since $\mathbf{y}_0$ is unstable under removal of a man.

We now turn to prove the fourth case. Here, H' is as given in Table 6. So now

$$2BER_{H'} = \frac{x_0 - 1 + v_0}{b_0} + \frac{u_1 + y_1}{b_1},$$

and, therefore,

$$2(BER_{H'} - BER_H) = \frac{x_0 - 1 - y_0}{b_0} + \frac{y_1 - x_1}{b_1}.$$

It is easy to see that the instability of $\mathbf{y}_0$ and $\mathbf{y}_1$ (i.e., the value of f_H flipped on both of those vectors due to the move of a single man from the former to the latter) implies that $x_0 = \frac{b_0 x_1}{b_1} + \delta + 1$ while $y_0 = \frac{b_0 y_1}{b_1} + \theta$, for some $\delta, \theta \in [-1, 0)$. Plugging this in the last equation reveals that $2(BER_{H'} - BER_H) = \frac{\eta}{b_0}$ where $\eta = \delta - \theta$ can take values in the interval $(-1, 1)$. □

A.4 Proof of Lemma 2

The proof of Lemma 2 is similar to that of Lemma 1 and thus omitted.

H	$Y_0^- := Y_0 \setminus \{\mathbf{y}_0\}$	$\mathbf{y}_0$	$\mathbf{y}_1$	$Y_1^- := Y_1 \setminus \{\mathbf{y}_1\}$
$\mathbf{b} = 0$	$u_0 := \sum_{\mathbf{y} \in Y_0^-} H(0, \mathbf{y})$	$x_0 := H(0, \mathbf{y}_0)$	$y_0 := H(0, \mathbf{y}_1)$#	$v_0 := \sum_{\mathbf{y} \in Y_1^-} H(0, \mathbf{y})$#
$\mathbf{b} = 1$	$u_1 := \sum_{\mathbf{y} \in Y_0^-} H(1, \mathbf{y})$#	$x_1 := H(1, \mathbf{y}_0)$#	$y_1 := H(1, \mathbf{y}_1)$	$v_1 := \sum_{\mathbf{y} \in Y_1^-} H(1, \mathbf{y})$

Table 3: The a-priori contingency table H

H'	$Y_0^- := Y_0 \setminus \{\mathbf{y}_0\}$	$\mathbf{y}_0$	$\mathbf{y}_1$	$Y_1^- := Y_1 \setminus \{\mathbf{y}_1\}$
$\mathbf{b} = 0$	u_0	$x_0 - 1$	$y_0 + 1$	v_0#
$\mathbf{b} = 1$	u_1#	x_1#	y_1#	v_1

Table 4: The a-posteriori contingency table H' in Case 2

H'	$Y_0^- := Y_0 \setminus \{\mathbf{y}_0\}$	$\mathbf{y}_0$	$\mathbf{y}_1$	$Y_1^- := Y_1 \setminus \{\mathbf{y}_1\}$
$\mathbf{b} = 0$	u_0	$x_0 - 1$#	$y_0 + 1$#	v_0#
$\mathbf{b} = 1$	u_1#	x_1	y_1	v_1

Table 5: The a-posteriori contingency table H' in Case 3

H'	$Y_0^- := Y_0 \setminus \{\mathbf{y}_0\}$	$\mathbf{y}_0$	$\mathbf{y}_1$	$Y_1^- := Y_1 \setminus \{\mathbf{y}_1\}$
$\mathbf{b} = 0$	u_0	$x_0 - 1$#	$y_0 + 1$	v_0#
$\mathbf{b} = 1$	u_1#	x_1	y_1#	v_1

Table 6: The a-posteriori contingency table H' in Case 4

A.5 Proof of Theorem 3

The first claim in the theorem follows directly from Lemma 1 (by considering each of the four possible cases in the lemma), while the second one follows similarly from Lemma 2. To see that, observe that a vector $\mathbf{y}_0$ is stable with respect to man-removals if $s(\mathbf{y}_0) \geq 1/b_0$ and stable with respect to woman-additions if $s(\mathbf{y}_0) \geq 1/b_1$; similarly, the vector $\mathbf{y}_1$ is stable with respect to man-additions if $s(\mathbf{y}_1) < -1/b_0$ and stable with respect to woman-removals if $s(\mathbf{y}_1) < -1/b_1$. □

A.6 Proof of Theorem 4

Let H (H') be the a-priori (a-posteriori) contingency table corresponding to C (C'). Let f_H and $f_{H'}$ be the corresponding MBPs. Denote by $Y_0 = f_H^{-1}(0)$ and $Y_1 = f_H^{-1}(1)$ the sets of vectors in Y that are mapped by the MBP f_H to 0 and 1, respectively. We assume that $Y_0, Y_1 \neq \emptyset$, since otherwise f_H would have been constant and then, by Theorem 2, already the initial BER would have equaled the maximal possible value of 1/2 and then C is the optimal solution (in which case $Y' = Y$).

Assume that there exists a vector $\mathbf{z} \in Y' \setminus Y$. Assume, without loss of generality, that $f_{H'}(\mathbf{z}) = 0$. Let $\mathbf{y}$ be any vector in Y_0. We proceed to prove that if we replace in C' all rows that equal $\mathbf{z}$ with the row $\mathbf{y}$, we will get a matrix C'' for which (a) the induced BER is the same as that of C', and (b) $\text{dist}(C, C'') \leq \text{dist}(C, C')$. Since C'' is an optimal solution, it implies that $\text{dist}(C, C'') = \text{dist}(C, C')$. Therefore, C'' is also an optimal solution and it does not include the row $\mathbf{z}$. By repeating the same argument for all rows in C'' which do not exist in the original C, we will arrive at an optimal solution matrix C^* in which the set of rows is included in the set of rows in C. That will conclude the proof.

We now prove claims (a) and (b) above. Denote

$$r_i = H'(i, \mathbf{y}) \,, \quad s_i = H'(i, \mathbf{z}) \,.$$

Since C' is an MBP-respecting solution and $\mathbf{y} \in Y_0$, then $\frac{r_0}{b_0} \geq \frac{r_1}{b_1}$. Also, since $f_{H'}(\mathbf{z}) = 0$, we have $\frac{s_0}{b_0} \geq \frac{s_1}{b_1}$. The overall contribution of those two columns to $BER(\mathbf{f}_{H'}, \mathbf{b})$ is $\frac{r_1 + s_1}{2b_1}$. In H'' (the contingency table of C''), those two columns will be merged into $\mathbf{y}$ and it is then clear that the contribution of the merged column to $BER(\mathbf{f}_{H''}, \mathbf{b})$ (where $f_{H''}$ is the corresponding MBP) would be still $\frac{r_1 + s_1}{2b_1}$, while the contribution of all other columns would remain unchanged. Hence, $BER(\mathbf{f}_{H''}, \mathbf{b}) = BER(\mathbf{f}_{H'}, \mathbf{b})$. That proves claim (a). Next, since the recommendation vector $\mathbf{z}$ does not exist as a row in C, its contribution to $\text{dist}(C, C')$ is $s_0 + s_1$. In the transition from C' to C'' we replace the recommendation vector of all $s_0 + s_1$ users that were offered $\mathbf{z}$ to $\mathbf{y}$. Since the latter vector does exist in C, we infer that its contribution to $\text{dist}(C, C'')$ is *at most* $s_0 + s_1$. Therefore, $\text{dist}(C, C'') \leq \text{dist}(C, C')$. Claim (b) is thus proved too. □

A.7 Proof of Lemma 3

By Theorem 3 and its proof we infer that: (a) for any $\mathbf{y} \in Y_0$, $\lfloor b_1 s(\mathbf{y}) \rfloor$ is the largest number of women whose recommendation vector can be changed from some vector in Y_1 to $\mathbf{y}$, without changing the value of f_H on $\mathbf{y}$; and (b) for any $\mathbf{y} \in Y_1$, $\lceil -b_1 s(\mathbf{y}) - 1 \rceil$ is the largest number of women whose recommendation vector can be changed from $\mathbf{y}$ to some vector in Y_0, without changing the value of f_H on $\mathbf{y}$.

Therefore, since any BER-increasing and MBP-respecting woman-move is of the form $(u, \mathbf{y}, \mathbf{z})$ where u is a woman, $\mathbf{y} \in Y_1$ is her original recommendation vector and $\mathbf{z} \in Y_0$ is her new recommendation vector, then the maximal number of such moves that are possible (while maintaining the MBP) is the minimum between the total number of women that the Y_0 vectors can "take in" and the total number of women that the Y_1 vectors can "loose" without affecting the MBP, as given in Eq. (5).

The proof of the corresponding claim for man-moves goes along the same lines. □

A.8 Proof of Theorem 5

Each matrix in $\mathcal{M}(C)$ is obtained from C by a sequence of moves which are MBP-respecting. Hence, as implied by Theorem 3, each such move increases the BER of the underlying MBP either by $1/2b_0$ (if the move is of a man) or by $1/2b_1$ (woman). Moreover, the order of moves in that sequence does not matter, since none of those moves changes the MBP and, consequently, none of those moves has an effect on the utility (in terms of the increase in the BER) of moves that are made later on. Hence, we may encode any solution in $\mathcal{M}(C)$ by two integers: n_0 (the number of man-moves) and n_1 (the number of woman-moves). We would like to stress that the pair (n_0, n_1) does not characterize the solution (since, for such a characterization it is needed to spell out the identity of the $n_0 + n_1$ affected users and their modified recommendation vectors). However, in the context of this proof, that pair is sufficient since it determines the final BER and also the resulting μ_1-distance. Indeed, if $C^* \in \mathcal{M}(C)$ is characterized by (n_0, n_1) then

$$BER^* = BER + \frac{n_0}{2b_0} + \frac{n_1}{2b_1}$$

(where BER and BER^* are the a-priori and a-posteriori BER values of the MBP) and

$$\mathrm{dist}(C, C^*) = n_0 + n_1 \,. \tag{9}$$

Next, $\mathcal{M}(C, \varepsilon)$ consists of all matrices in $\mathcal{M}(C)$ for which

$$\frac{n_0}{2b_0} + \frac{n_1}{2b_1} \geq \varepsilon - BER \,. \tag{10}$$

If that set is not empty then it is clear that an optimal solution is one in which n_1 is maximized (if $b_0 \geq b_1$) since an optimal solution minimizes the distance in Eq. (9) while still meeting the condition in Eq. (10). The solution issued by Algorithm FaiRecSys is indeed a solution that uses the maximal possible n_1 woman-moves. Hence, it is an optimal solution within the class $\mathcal{M}(C)$. If, on the other hand, the upper limits on n_0 and n_1, as dictated by Lemma 3, do not allow meeting the condition in Eq. (10), then it is clear that the practice implemented in Algorithm FaiRecSys of exhausting first all possible woman-moves results in the highest possible a-posteriori BER, δ. □

A.9 Proof of Lemma 4

Under our assumption that $b_0 \geq b_1$, the only MBP-improving and respecting moves are $(u, \mathbf{y}, \mathbf{z})$ where u is a man, $\mathbf{y} \in Y_0$ is his original recommendation vector, and $\mathbf{z} \in Y_1$ is an existing recommendation vector for a woman. Let $q \geq 1$ be the integer such that $b_0 = qb_1 + r$, $r \in [0, b_1)$. Then each column $\mathbf{z} \in Y_1$ can take at most q men without flipping the MBP's value on $\mathbf{y}$. Since $|Y_1| = b_1$ we may perform up to $b_1 q$ MBP-improving and respecting man-moves. As each such move yields a BER increase of $\frac{1}{2b_1}$, then after completing all of those moves the BER will increase from 0 to $\frac{b_1 q}{2b_0} = \frac{1}{2} \cdot \frac{b_0 - r}{b_0} = \frac{1}{2} \cdot \frac{qb_1}{qb_1 + r}$. Since $r \leq b_1 - 1$ and $q \geq 1$ we infer that we may increase BER to

$$\frac{1}{2} \cdot \frac{qb_1}{qb_1 + r} \geq \frac{1}{2} \cdot \frac{qb_1}{qb_1 + b_1 - 1} \geq \frac{1}{2} \cdot \frac{b_1}{b_1 + b_1 - 1} > \frac{1}{4} \,.$$

□