FEIR: Quantifying and Reducing Envy and Inferiority for Fair Recommendation of Limited Resources

Nan Li[1], Bo Kang[1], Jefrey Lijffijt[1] and Tijl De Bie[1]

[1]Ghent University, Ghent 9000, Belgium

Abstract

In settings such as e-recruitment and online dating, recommendation involves distributing limited opportunities, calling for novel approaches to quantify and enforce fairness. We introduce *inferiority*, a novel (un)fairness measure quantifying a user's competitive disadvantage for their recommended items. Inferiority complements *envy*, a fairness notion measuring preference for others' recommendations. We combine inferiority and envy with *utility*, an accuracy-related measure of aggregated relevancy scores. Since these measures are non-differentiable, we reformulate them using a probabilistic interpretation of recommender systems, yielding differentiable versions. We combine these loss functions in a multi-objective optimization problem called FEIR (Fairness through Envy and Inferiority Reduction), applied as post-processing for standard recommender systems. Experiments on synthetic and real-world data demonstrate that our approach improves trade-offs between inferiority, envy, and utility compared to naive recommendations and the baseline methods.

1. Introduction

Fairness in machine learning based recommendation systems attracts increasing research attention, driven both by ethical and legal motivations. Here we focus on recommending items with **limited availability**, such as job recommendation, online dating, and education resource recommendation. The need for users to *compete* for recommended items distinguishes this recommendation setting from more standard ones such as e-commerce, or movie or music recommendation, where items have practically unlimited availability.

When multiple users are recommended the same item, they enter a competition for that item. Only one or a few will win and obtain the item, leaving the others empty-handed. For example, a job seeker who applies for their recommended jobs could fail to get employed if these jobs were also recommended to better qualified rivals. This competition aspect brings specific challenges to evaluate and improve the fairness of recommendation strategies—challenges that have hitherto not been recognized.

To discuss this setting, it is useful to consider two possibly distinct kinds of affinity between a user and an item: an item's **utility** for the user (i.e. the user's preference), and a user's **suitability** (i.e. competitiveness) for the item. In traditional recommender systems, only utility is relevant, as suitability is directly related to the competitive nature of the setting.

We consider two ways in which unfairness can arise in such settings. First, a fair recommendation system for limited resources should ensure that users prefer their own recommended items over those recommended to others. This idea is captured by the notion of **envy**: user A has *envy* towards user B if the utility of user B's recommendations in user A's perspective is higher than user A's recommendations. Second, it is also arguably unfair to an individual if her recommended items are *always* also recommended to people more suitable to them than her, as she would fruitlessly compete for it. This idea is captured by the notion of **inferiority**: user A is *inferior* to user B if A is less suitable than B to the items recommended to both A and B. We argue that a fair recommender system in this setting should yield low envy and low inferiority for everyone.

As an illustration, consider the scenario of two users, **1** and **2**, and three items, ○, □, and △. Let the *utility* scores be represented by the matrix: $\begin{bmatrix} \bigcirc 0.2 & \square 0.6 & \triangle 0.9 \\ \bigcirc 0.1 & \square 0.8 & \triangle 0.7 \end{bmatrix}$, where the first row represents user **1**'s scores and the second row represents **2**'s scores. Similarly, let the *suitability* scores, or the chances of a user getting the item, be represented by the matrix: $\begin{bmatrix} \bigcirc 0.3 & \square 0.9 & \triangle 0.4 \\ \bigcirc 0.3 & \square 0.8 & \triangle 0.8 \end{bmatrix}$.

Examples: Recommending only △ to both users results in no envy, as the recommendations are equivalent and thus neither user prefers the other's. However, there is high inferiority, as **1** is less suitable than **2**, and thus less likely to obtain △. Recommending ○ to **1** and □ to **2** results in high envy as **2**'s recommendation has higher utility for **1** than their own recommendation, but no inferiority, as both users are recommended an item that is only recommended to themselves. Recommending ○ to both users results in neither envy, nor inferiority, but has low utility for both users. What is the best recommendation in this case depends on the chosen trade-off.

RecSys in HR'23: The 3rd Workshop on Recommender Systems for Human Resources, in conjunction with the 17th ACM Conference on Recommender Systems, September 18–22, 2023, Singapore, Singapore.

*Corresponding author.

nan.li@ugent.be (N. Li); bo.kang@ugent.be (B. Kang); jefrey.lijffijt@ugent.be (J. Lijffijt); tijl.debie@ugent.be (T. De Bie)

CEUR Workshop Proceedings (CEUR-WS.org)

The illustration above shows that both fairness notions are necessary: minimizing inferiority tends to result in less preferred jobs being recommended, which, if left uncontrolled, risks increasing envy. Moreover, there is also a trade-off between utility and both notions of fairness, particularly but not exclusively with inferiority.

Given the high stakes involved in many applications of this setting (with job recommendation as a notable example), there is an urgent need to adopt these notions of fairness in practical applications. While there is some work on the related notion of *congestion* and some limited work has been done on envy in recommender systems (see Sec. 4), we are unaware of any research directly addressing this need. This paper fills that gap, by formalizing these concepts as well as by proposing the FEIR (**F**airness through **E**nvy and **I**nferiority **R**eduction) method for post-processing the results of any other recommendation algorithm, yielding recommendations with low envy and low inferiority, while still maintaining high utility. Our specific contributions are:

1. We propose and formalize inferiority as a new individual fairness concept that is complementary to envy, when recommending items with limited availability. To facilitate minimizing these notions, we also derive their expected values with respect to a probabilistic interpretation of recommendation algorithms, resulting in differentiable versions. (Sec. 2.1.)

2. Leveraging these differentiable versions, we propose the FEIR algorithm, a model-agnostic post-processing method of the output scores of any upstream recommendation algorithm for all user-item pairs. FEIR seeks a fairer score matrix by solving a multi-objective optimization problem with the goal of minimizing the expected envy and inferiority, and maximizing the expected utility. (Sec. 2.2.)

3. We investigate FEIR's ability to trade-off both fairness measures and utility in extensive experiments both on synthetic and real data. We also demonstrate superiority of FEIR compared with the baseline methods. (Sec. 3.)

2. Method

In this section, we first give quantifications of utility, envy and inferiority in the *deterministic* setting and the *probabilistic* setting (Sec. 2.1). Second, we formulate the problem of finding a good recommendation strategy as a multi-objective optimization problem solvable by minimizing a weighted sum of loss terms, leading to the FEIR method (Sec. 2.2).

2.1. Quantification

Let $\boldsymbol{a} = (a_1, \dots, a_m)$ be m users, $\boldsymbol{b} = (b_1, \dots, b_n)$ be n items. A recommender system recommends k items to every user. The utility matrix $\boldsymbol{U}$ is an $m \times n$ matrix, where each entry $U_{i,j} \in (0, 1)$ represents the utility of item b_j to user a_i, so that each row $\boldsymbol{U}_{i,:}$ represents the utility function of a_i evaluated on all the items. The suitability matrix $\boldsymbol{S}$ is also an $m \times n$ matrix where each entry $S_{i,j} \in (0, 1)$ represents the suitability (matching degree), between user a_i and item b_j.

2.1.1. Deterministic setting

$\boldsymbol{U}$ gives us the item-wise utility, but in recommendation we need to measure the utility of a list of k items. Note that this list is a k-sized multiset constructed from $\boldsymbol{b}$ with *repeated recommendations allowed*, although in practice the chance of repetition is very slim when $k \ll n$. Another motivation of allowing repetition is for mathematical convenience as shown in 2.5.

Let $\boldsymbol{C}^k$ be an $m \times n$ counting matrix where each entry $C^k_{i,j} \in \mathbb{N}^0$ is the number of occurrences of job b_j in the recommendation for a_i. Then each row $\boldsymbol{C}^k_{i,:}$ represents the recommendation list for a_i such that $\sum_{j=1}^n C^k_{i,j} = k$ for all i. We omit the superscript k if the context is clear.

Definition 2.1 (User utility). *The **utility** of the recommendation for job seeker a_i is a simple summation of the utility of each job to a_i in a_i's list:*

$$u(a_i, \boldsymbol{U}, \boldsymbol{C}_{i,:}) = \sum_{j=1}^{n} U_{i,j} C_{i,j}.$$

Envy measures the comparative utility from each individual's perspective. It captures the idea that an individual may feel envy towards another if another person's recommended items have higher utility to them, wrt. their own utility.[1]

Definition 2.2 (User envy). *The **envy** from a_i to a_{i^*} is:*

$$e(a_i, a_{i^*}, \boldsymbol{U}, \boldsymbol{C}) = \sum_{j=1}^{n} U_{i,j}(C_{i^*,j} - C_{i,j}).$$

Inferiority represents the disadvantage of one user to another when they compete for the same items, such as applying for the same jobs. It is measured based on the suitability between users and items, represented by the matrix $\boldsymbol{S}$.

[1] If repetition is not allowed, the formulation would use an indicator function representing whether item j is recommended to user i. The major drawback of this formulation is that its probabilistic counterpart (the process of sampling without replacement) follows the hypergeometric distribution, which is computationally difficult. Further investigation is left future work.

Definition 2.3 (User inferiority)**.** *The **inferiority** from a_i to a_{i^*} is:*

$$f(a_i, a_{i^*}, \boldsymbol{S}, \boldsymbol{C}) = \sum_{j=1}^{n} \max(0, S_{i^*,j} - S_{i,j}) \cdot \min(1, C_{i,j} C_{i^*,j}).$$

Inferiority captures the difference in suitability between user a_i and a_{i^*} towards all the *common* recommended items, i.e., inferiority is only concerned with items recommended to both users. Note that, if any item occurred more than once, we only count it once, this is to consider the competition between them over the same item only once.

Definition 2.4. *Utility, envy, and inferiority on the* system *level are simply the averages of the positive user-level measurements:*

$$u(\boldsymbol{a}, \boldsymbol{U}, \boldsymbol{C}) = \tfrac{1}{m} \sum_{i=1}^{m} u(a_i, \boldsymbol{U}, \boldsymbol{C}), \tag{1}$$

$$e(\boldsymbol{a}, \boldsymbol{U}, \boldsymbol{C}) = \tfrac{1}{m} \sum_{1 \le i \ne i^* \le m} \max\left(0, e\left(a_i, a_{i^*}, \boldsymbol{U}, \boldsymbol{C}\right)\right), \tag{2}$$

$$f(\boldsymbol{a}, \boldsymbol{S}, \boldsymbol{C}) = \tfrac{1}{m} \sum_{1 \le i \ne i^* \le m} f(a_i, a_{i^*}, \boldsymbol{S}, \boldsymbol{C}). \tag{3}$$

The $\max(0, \cdot)$ in the definition of the envy ensures that only positive contributions are counted, to avoid a negative envy in one user to compensate a positive envy in another. (Individual user utilities and inferiorities are always positive.)

2.1.2. Probabilistic setting

The discontinous nature of recommender systems as recommending multisets of items makes it practically impossible to utilize the utility, envy, and inferiority from Def. 2.4 in an optimization-based approach. We will thus develop probabilistic alternatives that are differentiable.

We consider a probabilistic recommendation setting in which the recommendation strategy is represented by a user-item mapping function, $\pi : \boldsymbol{a} \times \boldsymbol{b} \to [0, 1]$, that assigns a probability to each user-item pair of the recommendation of the item to the user. To model the recommendations, we assume an independent multinomial process for each user, where each user has a different n-sided uneven dice, and to recommend k ($k \ll n$) items, we throw the dice k times and take the outcome as the recommendation.

Then a recommender strategy can be represented as an $m \times n$ matrix $\boldsymbol{P} \in [0, 1]^{m \times n}$ where each entry $P_{i,j}$ is the probability of recommending b_j to a_i. All users' k-sized recommendation can be written as a random matrix $\mathbf{X}$ where each row $\mathbf{X}_{i,:}$ is a random vector for a_i where the random variables $\mathrm{X}_{i,j}$ indicate the number of times item b_j is included in a_i's list. By our setting $\mathbf{X}_{i,:}$ follows a multinomial distribution with parameters k and $\boldsymbol{P}_{i,:}$.

In this context of probabilistic recommendation, the *expected values* of a user's utility, envy, and inferiority are given by the following Proposition:

Proposition 2.5 (Expected user utility, envy, and inferiority)**.**

$$\mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[u(a_i, \boldsymbol{U}, \mathbf{X}_{i,:})] = k \sum_{j=1}^{n} P_{i,j} U_{i,j}, \tag{4}$$

$$\mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[e(a_i, a_{i^*}, \boldsymbol{U}, \mathbf{X})] = k \sum_{j=1}^{n} (P_{i^*,j} - P_{i,j}) U_{i,j}, \tag{5}$$

$$\begin{aligned}\mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}&[f(a_i, a_{i^*}, \boldsymbol{S}, \mathbf{X})] \\ &= \sum_{j=1}^{n} \max(0, S_{i^*,j} - S_{i,j}) \\ &\quad \cdot (1 - (1 - P_{i,j})^k)(1 - (1 - P_{i^*,j})^k).\end{aligned} \tag{6}$$

Proof outline. For utility and envy, this follows from the fact that $\mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[X_{i.j}] = k P_{i,j}$ (the factor k stemming from $\sum_{i=1}^{n} x_i = k$), and from linearity of the expectation operator. For inferiority, this follows from linearity of the expectation operator, and from the fact that $(1-(1-P_{i,j})^k)(1-(1-P_{i^*,j})^k)$ is the probability that both $C_{i,j}$ and $C_{i*,j}$ are non-zero integers, and thus the probability that $\min(1, C_{i,j} C_{i*,j})$ is equal to 1. □

For a recommendation system for users $\boldsymbol{a}$ over items $\boldsymbol{b}$, represented by the $m \times n$ matrix $\boldsymbol{P}$, the expected utility, envy and inferiority of the k-sized recommendation $\mathbf{X}$ on the *system* level is the average of the expected values of all users:

$$\mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[u(\boldsymbol{a}, \boldsymbol{U}, \mathbf{X})] = \tfrac{1}{m} \textstyle\sum_{i=1}^{m} \mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[u(a_i, \boldsymbol{U}, \mathbf{X}_{i,:})], \tag{7}$$

$$\mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[e(\boldsymbol{a}, \boldsymbol{U}, \mathbf{X})] = \tfrac{1}{m} \textstyle\sum_{1 \le i \ne i^* \le m} \max(0, \mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[e(a_i, a_{i^*}, \boldsymbol{U}, \mathbf{X})]), \tag{8}$$

$$\mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[f(\boldsymbol{a}, \boldsymbol{S}, \mathbf{X})] = \tfrac{1}{m} \textstyle\sum_{1 \le i \ne i^* \le m} \mathbb{E}_{\mathbf{X} \sim \boldsymbol{P}}[f(a_i, a_{i^*}, \boldsymbol{S}, \mathbf{X})]. \tag{9}$$

2.2. Optimization by minimizing combined losses

The FEIR algorithm minimizes a combined loss function defined from the expected utility, inferiority, and envy of the recommendation system. It uses a gradient descent based method to optimize the scores of the resulting recommendation strategy, represented by the matrix $\boldsymbol{P}'$, by solving:

$$\begin{aligned}\ell_{total}(\boldsymbol{P}', \boldsymbol{S}, \boldsymbol{U}) = w_1 \ell_e(\boldsymbol{P}', \boldsymbol{U}) \\ + w_2 \ell_f(\boldsymbol{P}', \boldsymbol{S}) \\ + w_3 \ell_u(\boldsymbol{P}', \boldsymbol{U}) \\ + w_4 \ell_p(\boldsymbol{P}'),\end{aligned} \tag{10}$$

where $\ell_e(\boldsymbol{P}', \boldsymbol{U})$, $\ell_f(\boldsymbol{P}', \boldsymbol{S})$, $\ell_u(\boldsymbol{P}', \boldsymbol{U})$, and $\ell_p(\boldsymbol{P}') = \sum_{i=1}^{m}(\sum_{j=1}^{n} P'_{i,j} - 1)^2$ are the expected envy (Eq. 8), expected inferiority (Eq. 9), negative expected utility (Eq. 7), and a penalty term for making each row of $\boldsymbol{P}'$ a probability distribution, respectively. Parameters w_1, w_2, w_3, w_4 are weights for each term. The penalty term $\ell_p(\boldsymbol{P}')$ can be omitted if the matrix $\boldsymbol{P}'$ is renormalized after each update (e.g., using row-wise softmax as the activation function). An important benefit of FEIR is its being model-agnostic: any model capable of scoring all user-item pairs can be post-processed by FEIR.

Notes on probabilistic and deterministic settings. The probabilistic setting is more general and mathematically convenient, but real recommendation systems typically recommend (deterministically) each user the k items with the highest probabilities. Thus, our experiments train with the probabilistic but evaluate with the deterministic measures.

Notes on available affinity types in current systems. In our definitions, $\boldsymbol{S}$ represents the suitability of users to items, and $\boldsymbol{U}$ quantifies the utility of items to users. The difference between them can create tension between envy and inferiority, as seen when a job seeker prefers unsuitable jobs. However, real-world applications typically use a single affinity score provided by an existing recommender system, combining both suitability and utility. Thus, for practical reasons, in our experiments only one set of affinities is used to calculate both envy and inferiority, except for one synthetic dataset. In these cases, utilities and suitabilities align, but tension between envy and inferiority still arises due to individual differences in scores. This is illustrated by a toy example where the scores for users **1** and **2** with respect to items are $\begin{bmatrix} \bigcirc 0.1 & \square 0.9 & \triangle 0.8 \\ \bigcirc 0.4 & \square 0.6 & \triangle 0.5 \end{bmatrix}$, and recommending □ to both of them results in no envy and high utility, but high inferiority from **2** to **1**.

2.3. Scaling-up methods

With large scale data, we propose the following approximation methods: inferiority loss mini-batching, user sampling, item sampling, user-item sampling. *Mini-batching* randomly splits the m users into $\lfloor m/b \rfloor$ batches and at each step calculates the inferiority from b users within the current mini-batch to all the users with respect to all the items, leaving other losses calculated globally. *User sampling* takes a random subset of m_s users at each training step and calculates the losses within this subset. *Item sampling* takes a random subset of n_s items at each training step and calculates the losses between all user pairs with respect to only those items. *User-item sampling* samples from both the users and items at each training step.

2.4. Metrics

We evaluate recommendation strategies based on the one-time deterministic recommendation obtained from the probabilistic strategy. For different ks, let $\boldsymbol{C}^k$ be the binary matrix obtained from the recommendation strategy $\boldsymbol{P}$ by setting the item indices with the highest k values in each row to be 1 and the rest to be 0.

Normalized system-level utility and fairness. The system-level utility, envy and inferiority for top-k recommendation are defined by Eq. 1, 2 and 3, albeit $\boldsymbol{U} = \boldsymbol{S}$ in the experimental data. We also calculate the **overall fairness** as

$$g(\boldsymbol{a}, \boldsymbol{U}, \boldsymbol{S}, \boldsymbol{C}^k) = e(\boldsymbol{a}, \boldsymbol{U}, \boldsymbol{C}^k) + f(\boldsymbol{a}, \boldsymbol{S}, \boldsymbol{C}^k).$$

Let C^K_{naive} denote the naive recommendation, then we have the **normalized system-level top-k recommendation metrics** defined as

$$\frac{\psi(\boldsymbol{a}, \boldsymbol{U}, \boldsymbol{C}^k)}{\psi(\boldsymbol{a}, \boldsymbol{U}, \boldsymbol{C}^k_{naive})},$$

where $\psi \in \{u, f, g\}$ (no normalized envy since $e(\boldsymbol{a}, \boldsymbol{U}, \boldsymbol{C}^k_{naive}) = 0$).

Competition faced by each user. To address RQ2, we use the following competition indicators. The **mean rank** of job seeker a_i is calculated as:

$$\text{rank}(i) := \frac{1}{k}\sum_{j=1}^{n} |D_{i,j}|,$$

where

$$D_{i,j} = \{a_{i^*} | C_{i,j} = C_{i^*,j} = 1, S_{i^*,j} > S_{i,j}\},$$

which measures the average rank of user a_i among her competitors for the same recommended items. The **mean suitability gap** of user a_i is calculated as:

$$\text{gap}(i) = \frac{1}{k}\sum_{j=1}^{n} C_{i,j} \frac{1}{\max(1, |D_{i,j}|)} \sum_{i^* \in D_{i,j}} (S_{i^*,j} - S_{i,j}),$$

which measures the average difference in suitability scores between user a_i and her likelier competitors for the same recommended items. By averaging these metrics over all users, we can obtain an overall evaluation of the competition.

Multiple solutions comparison. For methods that can generate multiple solutions representing different levels of trade-offs, we plot the Pareto frontiers to visually compare sets of solutions. Additionally, we use the following numerical metrics:

1. **HV** (hypervolume) measures the amount of the objective space (relative to a reference point) that is dominated by the points on the frontier.
2. **Fairness above utility threshold** $\min(\phi|t)$: the minimum value of a fairness metric ϕ among all solutions with utility higher than t, where ϕ could be inferiority, overall fairness, mean rank or mean suitability gap. This allows us to assess how well a solution performs in terms of fairness for a given level of utility.

Item-side fairness. Although our focus is user-side fairness, we also investigated the item-side fairness by comparing the Gini index of item exposure ([1, 2, 3, 4]) before and after FEIR post-processing.

3. Experiments

To evaluate the effectiveness of FEIR, we conduct experiments to answer the following research questions:

RQ1. How does FEIR compare to the baseline methods in improving the *trade-offs between envy, inferiority and utility*?

RQ2. Does FEIR decrease the *competition measurements defined from rivals* compared to the baseline methods?

3.1. Datasets

In our experiments, we use a variety of synthetic and real-world datasets to evaluate the performance of our proposed method. There are three types of synthetic datasets. **Random synthetic data with distinct suitability and utility (SU50)**: Two 50×50 real-numbered matrices generated from a truncated normal distribution $(0, 1)$ representing suitability scores and utility scores for 50 users and 50 items. **Random synthetic data with one set of scores**: Matrices generated from a truncated normal distribution $(0, 1)$ with *varying ratios* of number of users and items to investigate the effect of varying these ratios. **Structured synthetic data with one set of scores**: Two 20×100 real-numbered matrices that simulate specific scenarios: Item groups (IG) and User groups (UG). The **IG** dataset represents the scenario where certain items have generally higher scores across all users, while the **UG** dataset represents the scenario where certain users have generally higher scores across all items.

We also use four real-world datasets, all obtained from the same upstream job recommendation model based on [5]. **Zhilian**: Scores for 2,781 users and 6,568 items, sampled from a public dataset provided by a Chinese online recruitment platform. **CareerBuilder**: Scores for 7,459 users and 11,020 items, obtained from a public dataset provided by *CareerBuilder*. **VDAB small**: Scores for 1,186 users and 8,921 items, a random sample from a private dataset provided by a labor agency in Belgium. **VDAB large**: Scores for 10,369 users and 66,898 items, a random sample from the same source as VDAB small, but including more data. Results of this dataset are omitted due to space limitation.[2]

3.2. Baselines

We use the following four baseline methods.

Standard recommendation (Naive). Given the scores between all users and items, the common practice is to recommend the items with the highest k scores to each user.

Randomization of top scored items (Shuffle). Randomly Sample k items from items with the top-d ($\geq k$) scores.

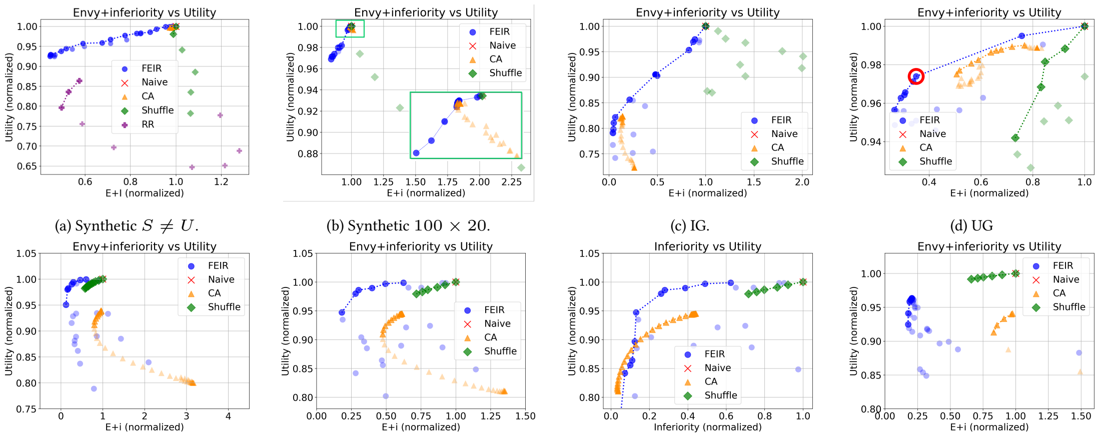
Congestion alleviation method (CA). Naya et al. [6] proposes a congestion alleviation method based on linear optimization that aims to decreases the competition in the job market by using optimal transport. CA casts the problem of minimizing congestion into a linear program where the objective is to maximize the element-wise product of the original probability matrix and the solution matrix under the constraint of evenly distributing the probability of recommending each item.

Modified Round-Robin procedure (RR). Modified based on [7], RR sets a threshold τ for suitability, randomly orders the users and then in each round, allocates one item for each user at each round such that this item is the most preferred one for this user with suitability greater than τ. k rounds would be run for top-k recommendation. Unlike the other methods, RR is applicable only when U and S are both available.

3.3. Experiment setting

For our method FEIR, we initialize the parameters by applying a row-wise softmax to the given scores and use gradient descent based methods to minimize the loss function defined in Eq. 10. We perform a coarse search to find an appropriate learning rate, and then use this value to train the model with different combinations of loss weights to achieve different trade-offs between envy, inferiority and utility. For the CA baseline, different entropic relaxation terms are used to roughly controls the

[2] Due to the size of **VDAB large**, we experimented several methods for scaling up, including sampling and mini-batching. The results are included in our online supplementary materials.



(a) Synthetic $S \neq U$.

(b) Synthetic 100×20.

(c) IG.

(d) UG

(e) VDAB small $k = 5$.

(f) VDAB small $k = 50$.

(g) VDAB small: Inferiority vs utility $k = 50$.

(h) VDAB large $k = 20$.

Figure 1: Selected Pareto frontiers trading-off envy, inferiority and utility (upper-left better). (b): upper right region zoomed in. (d): The circled FEIR solution decreases inferiority of **both** user groups from Naive: the advantageous group **0.094** → **0.082** and the other **4.251** → **1.152**. (h): user-item sampling with a sample size about $\frac{1}{30}$ of the total users and $\frac{1}{70}$ items.

trade-offs. For the synthetic datasets, we train and evaluate strategies for the top 10 recommendation. For the real-world datasets, we train and evaluate strategies for different ks, ranging from 1 to 100. For the VDAB large dataset only a medium size $k = 20$ is trained and evaluated due to time limitations.

We explore all scaling-up methods with the VDAB small dataset with $k = 100$, find all methods perform similarly besides item sampling. Therefore, we apply one method to each real-world dataset for a full range of ks: mini-batching to the VDAB small dataset, user sampling to the Zhilian and Careerbuilder datasets, user-item sampling to VDAB large.

3.4. Results

3.4.1. Fairness versus utility trade-offs (RQ1)

Our proposed method, FEIR, and the baseline methods were evaluated on synthetic and real-world datasets. The results indicate that both FEIR and CA can consistently improve fairness over the naive recommendation approach, while sacrificing some utility. By varying the hyperparameters for the methods, different trade-offs between fairness and utility were achieved. To compare the results, we plotted each solution as a point on a graph with (un)fairness as the x-coordinate and utility as the y-coordinate, and drew the Pareto frontiers.

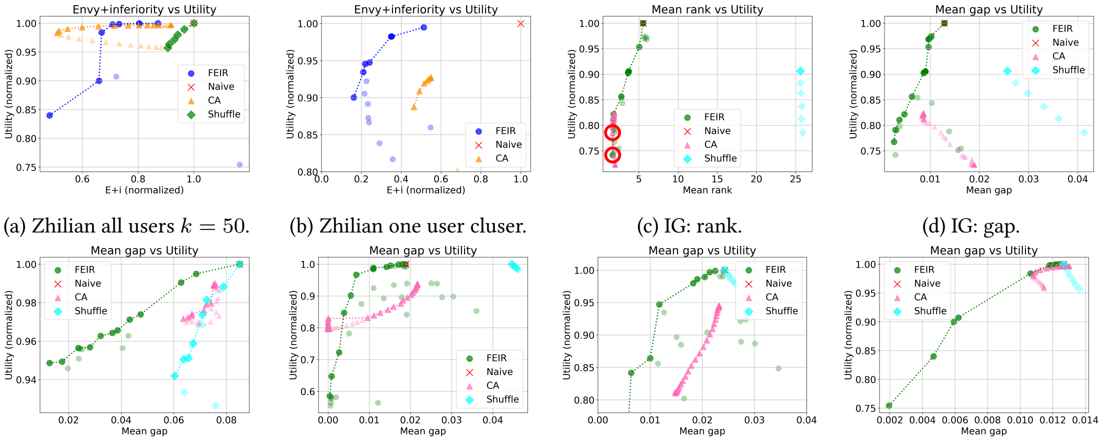
Synthetic datasets. FEIR is clearly the best (Fig. 1c, 1d), followed by CA, although the latter tends to cover a smaller solution region. RR scarifies too much utility for fairness (Fig. 1a). Shuffle performs unstably.

Interestingly, a closer look at one of our solutions for UG shows that FEIR can simultaneously decrease the inferiority for both user groups (Fig. 1d), which is desirable as it does not require sacrifices from one group to benefit the other.

When recommending items using the naive recommendation strategy with the random *synthetic datasets with varying user-item ratios*, the inferiority increases with an increased ratio of users to items, indicating that the naive approach causes competitive disadvantages for users, and the more limitation the tenser the competition. CA does not decrease inferiority well when the number of items is not greater than the number of users; on the other hand, FEIR is able to find solutions with low inferiority as seen in Fig. 1b. When the number of items surpasses users, CA can also find solutions with low inferiority and high utility, but is still outperformed by FEIR (corresponding figures included in our online supplementary.).

Real world datasets. Data exploration confirms the existence of inferiority and competition caused by the naive recommendation. With increasing ks, the utility per recommendation decreases, and the inferiority and competition increase with a decelerating growth rate (see figures in our online supplementary). The reason is that with a larger k, there are more overlapping recommendation and more competition, but also the average scores decrease with increasing k.

The *VDAB small* and *CareerBuilder* datasets show similar patterns in the relative performance of FEIR and CA. FEIR can decrease inferiority without reducing much utility or increasing envy, while CA decreases inferiority but also increases envy and reduces utility, especially



(a) Zhilian all users $k = 50$. (b) Zhilian one user cluser. (c) IG: rank. (d) IG: gap.

(e) UG. (f) VDAB small $k = 5$. (g) VDAB small $k = 50$. (h) Zhilian $k = 50$.

Figure 2: Compare (a) with (b): FEIR's performance on Zhilian dataset is not ideal when trained with *user sampling*, but FEIR outperforms CA when trained without sampling. (c)-(h): Selected Pareto frontiers trading off competition and utility (upper-left better). (c): Two solutions with close mean ranks are circled. The mean suitability gap of FEIR is **0.003** while CA's is **0.012**.

Table 1
Comparison of the Pareto frontiers trading off fairness metrics with utility for the **CareerBuilder** dataset with varying ks. The reference point for calculating the HVs is [1, 0.95]. The better results are marked bold.

k	HV(g vs u)		HV(i vs u)		min(g|0.95)		min(i|0.95)	
	FEIR	CA	FEIR	CA	FEIR	CA	FEIR	CA
1	**0.043**	0.013	**0.048**	0.031	**0.140**	0.642	**0.006**	**0.006**
5	**0.042**	0.024	**0.045**	0.031	**0.138**	0.365	**0.049**	0.127
10	**0.041**	0.026	**0.044**	0.032	**0.143**	0.321	**0.081**	0.142
20	**0.039**	0.027	**0.042**	0.031	**0.185**	0.321	**0.104**	0.175
50	**0.034**	0.026	**0.035**	0.029	**0.278**	0.357	**0.248**	**0.248**
100	**0.029**	0.025	**0.030**	0.027	**0.367**	0.392	0.333	**0.324**

when the number of recommendations is small. Shuffle prioritize utility, but cannot reduce much unfairness (Fig. 1e and 1f).

With large $k = 50$, FEIR's performance in reducing unfairness is not as good as CA for the *Zhilian* dataset when user sampling is used (Fig. 2a). Nonetheless, FEIR performs better than CA when trained on smaller subsets of users that can be processed in a single batch as seen in Fig. 2b. This suggests that the loss functions are effective, but the decreased performance is most likely due to the optimization process or some unique characteristics of the Zhilian dataset, which is left for future work.

FEIR performed well on the *VDAB large* dataset, even with a sample size relatively small to the total numbers as show in Fig. 1h.

Quantitative comparisons of the Pareto frontiers generated by FEIR and CA for the VDAB small, Zhilian and CareerBuilder datasets with various k values support these observations. We only present the results for *CareerBuilder* dataset here in Table 1 due to space limitation.

3.4.2. Competition faced by users (RQ2)

In general, CA is capable of achieving a low mean rank (Fig. 2c), but always a much higher mean gap compared to FEIR (Fig. 2d, 2e). We argue that FEIR is more desirable. A recommendation with a low mean rank but a large mean suitability gap suggests that, although a user does not have many competitors, the competitors she does have are much better hence much more likely to defeat this user. For example, consider a job seeker a_i with a suitability score of 0.7 for a certain job. CA tends to recommend this jobs to only one other job seeker with a score of 0.99, and on the other hand FEIR may recommend this job to three other job seekers with scoring 0.69, 0.74, 0.8 respectively. It is reasonable to believe that FEIR gives user a_i a better chance of getting hired, especially when considering that in reality, one would not apply for all recommended jobs. Shuffle performs almost always the worst.

When recommending a small number of jobs from a large pool, CA sometimes recommends non-overlapping jobs to each user, resulting in trivial solutions with no competition but decreased utility, as seen in the left most region of Fig. 2f. However, FEIR can provide solutions with higher utility. As k increases, it becomes harder to give non-overlapping recommendations for CA such that FEIR always gives a lower suitability gap (Fig. 2g, 2h).

A quantitative comparison of the Pareto frontiers gen-

Table 2
Comparison of the Pareto frontiers trading off competition metrics with utility for the **VDAB small** data with varying ks. The reference point for calculating the HV(rank vs u) being [50, 0.9] means the reference value of the mean rank is 50 and the normalized utility 0.9, and for HV(gap vs u) [0.03, 0.9] means the reference value of the mean suitability gap is 0.03.

k	HV(rank vs u)		HV(gap vs u)		$\min(\text{rank}\|0.9)$		$\min(\text{gap}\|0.9)$	
	FEIR	CA	FEIR	CA	FEIR	CA	FEIR	CA
1	**4.747**	1.496	**0.003**	0.0	**1.286**	3.66	**0.002**	0.017
5	**4.415**	1.54	**0.002**	0.0	**2.731**	5.318	**0.006**	0.02
10	**4.103**	1.535	**0.002**	0.0	**4.599**	6.761	**0.007**	0.021
20	**3.592**	1.523	**0.002**	0.0	**8.317**	8.805	**0.007**	0.022
50	**2.463**	1.438	**0.001**	0.0	15.462	**11.28**	**0.012**	0.021
100	**1.482**	1.291	**0.001**	0.0	29.442	**16.099**	**0.018**	0.022

erated by FEIR and CA for the *VDAB small* dataset with various k values shows that FEIR is better than CA almost across the board, except for $\min(\text{rank}|0.9)$ with $k = 50$ and 100 (Table 2). The *CareerBuilder* dataset has similar results with VDAB small where FEIR is better than CA in general, while FEIR shows less advantage over CA for *Zhilian* (plots and tables in our online supplementary), as discussed in Section 3.4.1.

3.4.3. Item-side fairness

FEIR improves the fairness to the items as the Gini index decreased greatly for all datasets after FEIR post-processing (Table 3).

Our *code and supplementary* materials for more details and extra plots are publicly available at https://github.com/aida-ugent/FEIR.

4. Related work

This paper extends the growing literature on fairness in machine learning (e.g. [8, 9, 10, 11, 12, 13, 14, 15, 16, 7]). Here we summarize the most directly related research.

Fairness when recommending items with limited availability. Particularly in the context of job recommendations, this is an increasingly active research area. Yet, the current literature mainly focuses on *group level* disparity notions. For example, Geyik et al. [17] proposed four deterministic reranking algorithms to mitigate biased prediction towards any sensitive job seeker group, and Islam et al. [18] addressed gender bias in job recommendations by proposing a neural fair collaborative filtering model. In contrast to this existing work, we focus on fairness from the perspective of *individual* users, rather than group level fairness. Other orthogonal research includes fairness for jobs and interdisciplinary studies (see recent survey by Mashayekhi et al. [19]).

Table 3
FEIR also improves the item-side fairness as the Gini indices of item exposure for all datasets are decreased after FEIR post-processing.

Dataset	IG	UG	V(S)	V(L)	ZL	CB
Gini index ↓ %	73	60	34	53	60	37

Competition and congestion in recommendation. To the best of our knowledge, there has been no research at all on the concept of inferiority. Yet, Naya et al. [6] did study the related notion of *congestion*, in the context of labor market. They proposed a congestion alleviation method, which reduces the intersection between the sets of jobs recommended to different job seekers. Congestion does not consider suitability (i.e. competitiveness) of users for their recommended jobs like inferiority does.

Envy-freeness in recommendation. Inspired by the literature on social choice theory and fair resource allocation (e.g., [20, 21, 22]), a few researchers recently introduced the notion of envy-freeness into the context of recommendation systems. Do et al. [23] gave a generic individual-level definition of envy-freeness and cast the problem of auditing for such envy-freeness as an exploration problem in multi-armed bandits. Their focus is online *evaluation* (auditing) of existing systems, while we aim to also *minimize* envy as well as inferiority, using a post-processing method. Patro et al. [7] designed a modified Round-Robin algorithm to ensure fairness on the item side while guaranteeing envy-freeness up to one good (EF1) fairness for every user, and Wu et al. [24] extended this approach to producer fairness. Besides the fact that we do not share their focus on item-side fairness, their problem settings do not apply to limited resource recommendation because the users in their setting do not compete with each other.

5. Discussion and Conclusion

Recommending items with limited availability to users has its own challenges and brings new fairness requirements not addressed in the existing literature. In this paper we proposed envy and inferiority as important fairness notions to fill the gap and presented a post-processing approach FEIR to improve the fairness of such recommendation settings.

Our experiments on synthetic and real job recommendation datasets demonstrated that FEIR improves fairness by reducing the potential competitive disadvantage of users without significantly sacrificing utility. Importantly, our method FEIR is not limited to the labor market, but also promising in reducing user inferiority and competitive disadvantages in other real-world scenarios such as online dating, paper bidding systems, and education resources recommendation.

Our work has limitations but also opens up new research opportunities. The actual competition and chances of getting any item depend on many factors beyond any recommendation system and hence beyond our scope. Also, emphasizing envy and inferiority does not make other existing fairness concerns any less important, nor the case that they can cover all new fairness requirements from the unique features of recommending limited resources. Rather, our findings create new opportunities for research to explore the relations among different fairness notions and identify other ignored dimensions of fairness in these settings.

Some alternative formulations of utility, envy and inferiority are possible. For example, disallowing repeated recommendation for a user, which involves further complexity in the probabilistic setting. It is also possible to modify the quantification of inferiority by taking the utility into account. The analysis and comparison of the current formulation and the alternatives would be interesting for future work. Besides, the interests of recruiters could be further considered by adapting the optimization objective to include some metrics representing the suitability of candidates. The dynamics between job seeker side and recruiter side fairness is another future direction worth exploring.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) (ERC Grant Agreement no. 615517), and under the European Union's Horizon 2020 research and innovation programme (ERC Grant Agreement no. 963924), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, and from the FWO (project no. G0F9816N, 3G042220). Part of the experiments were conducted on pseudonimized HR data generously provided by VDAB (Vlaamse Dienst voor Arbeidsbemiddeling en Beroepsopleiding).

References

[1] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, R. Burke, Fairmatch: A graph-based approach for improving aggregate diversity in recommender systems, 2020.

[2] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, R. Burke, A graph-based approach for mitigating multi-sided exposure bias in recommender systems, ACM Transactions on Information Systems 40 (2021) 1–31.

[3] Y. Ge, S. Liu, R. Gao, Y. Xian, Y. Li, X. Zhao, C. Pei, F. Sun, J. Ge, W. Ou, Y. Zhang, Towards long-term fairness in recommendation, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, ACM, 2021, pp. 445–453.

[4] V. Do, N. Usunier, Optimizing generalized gini indices for fairness in rankings, 2022.

[5] B. Kang, J. Lijffijt, T. De Bie, Conditional network embeddings, stat 1050 (2018) 22.

[6] V. Naya, G. Bied, P. Caillou, B. Crépon, C. Gaillac, E. Pérennes, M. Sebag, Designing labor market recommender systems: The importance of job seeker preferences and competition, 2021.

[7] G. K. Patro, A. Biswas, N. Ganguly, K. P. Gummadi, A. Chakraborty, Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms, in: Proceedings of The Web Conference 2020, 2020, pp. 1194–1204.

[8] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012, pp. 214–226.

[9] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, Advances in neural information processing systems 29 (2016) 3315–3323.

[10] M. B. Zafar, I. Valera, M. G. Rodriguez, K. P. Gummadi, A. Weller, From Parity to Preference-based Notions of Fairness in Classification, arXiv:1707.00010 [cs, stat] (2017).

[11] M. J. Kusner, J. R. Loftus, C. Russell, R. Silva, Counterfactual fairness, in: Proc. of NeurIPS, 2017, pp. 4069 – 4079.

[12] S. Yao, B. Huang, Beyond Parity: Fairness Objectives for Collaborative Filtering, arXiv:1705.08804 [cs, stat] (2017).

[13] H. Steck, Calibrated recommendations, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 154–162.

[14] L. Wang, T. Joachims, User Fairness, Item Fairness, and Diversity for Rankings in Two-Sided Markets, in: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, Association for Computing Machinery, New York, NY, USA, 2021, pp. 23–41.

[15] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Multistakeholder recommendation: Survey and research directions, User Modeling and User-Adapted Interaction 30 (2020) 127–158.

[16] V. Do, S. Corbett-Davies, J. Atif, N. Usunier, Two-sided fairness in rankings via Lorenz dominance, in: Advances in Neural Information Processing Systems, 2021, pp. 8596 – 8608.

[17] S. C. Geyik, S. Ambler, K. Kenthapadi, Fairness-aware ranking in search & recommendation systems with application to linkedin talent search, in: Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining, 2019, pp. 2221–2231.

[18] R. Islam, K. N. Keya, Z. Zeng, S. Pan, J. Foulds, Debiasing career recommendations with neural fair collaborative filtering, in: Proceedings of the Web Conference 2021, 2021, pp. 3779–3790.

[19] Y. Mashayekhi, N. Li, B. Kang, J. Lijffijt, T. De Bie, A challenge-based survey of e-recruitment recommendation systems, 2022.

[20] D. K. Foley, Resource allocation and the public sector, Yale economic essays 7 (1967).

[21] H. Moulin, Fair Division and Collective Welfare, The MIT Press, 2003.

[22] H. R. Varian, Equity, envy, and efficiency, Journal of Economic Theory 9 (1974) 63–91.

[23] V. Do, S. Corbett-Davies, J. Atif, N. Usunier, Online certification of preference-based fairness for personalized recommender systems, arXiv:2104.14527 [cs, stat] (2022).

[24] Y. Wu, J. Cao, G. Xu, Y. Tan, Tfrom: A two-sided fairness-aware recommendation model for both customers and providers, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 1013–1022.