FairRoad: Achieving Fairness for Recommender Systems with Optimized Antidote Data

Minghong Fang
The Ohio State University

Michinari Momma*
Amazon

Jia Liu*
The Ohio State University

Yi Sun*
Amazon

ABSTRACT

Today, recommender systems have played an increasingly important role in shaping our experiences of digital environments and social interactions. However, as recommender systems become ubiquitous in our society, recent years have also witnessed significant fairness concerns for recommender systems. Specifically, studies have shown that recommender systems may inherit or even amplify biases from historical data, and as a result, provide unfair recommendations. To address fairness risks in recommender systems, most of the previous approaches to date are focused on modifying either the existing training data samples or the deployed recommender algorithms, but unfortunately with limited degrees of success. In this paper, we propose a new approach called fair recommendation with optimized antidote data (FairRoad), which aims to improve the fairness performances of recommender systems through the construction of a small and carefully crafted antidote dataset. Toward this end, we formulate our antidote data generation task as a mathematical optimization problem, which minimizes the unfairness of the targeted recommender systems while not disrupting the deployed recommendation algorithms. Extensive experiments show that our proposed antidote data generation algorithm significantly improve the fairness of recommender systems with a small amounts of antidote data.

1 INTRODUCTION

Background and Motivation: Today, we are interacting with a large variety of recommender systems on a daily basis, including but not limited to finding accommodation (Airbnb and Booking.com), online video streaming (YouTube and DailyMotion), online music streaming (Spotify, Pandora, and SoundCloud), ride-sharing (Uber and Lyft), online shopping (eBay), Apps searching (App Store and Google Play Store), job searching (LinkedIn and Indeed), and even in some high-risk decision making settings, such as financial loan approval, hiring, etc. [43]. Digital environments and social interactions have fundamentally changed our social life by facilitating efficient information retrieval and context-aware searching and ranking results. As such, recommender systems have played an increasingly important role in shaping our experiences of digital environments and social interactions.

However, despite all the above salient features of recommender systems, recent studies have found that recommender systems could provide unfair and/or biased recommendations to users if not designed and maintained appropriately [69, 78]. For instance, an investigation on more than one million ratings in MovieLens dataset [63] has showed that, over time, the algorithm amplified the initial bias toward males and recommendations are deviated from female users’ preferences. Moreover, as a result of the deviation, the dataset is gradually dominated by ratings from males and losing its female population. As another example, it has been reported [2, 47] that job ads on Facebook for supermarket cashier positions were shown to 85% women, and job positions in Taxi companies were shown to audience that was almost 75% black.

To date, there have been some existing works attempting to address the fairness risks in recommender systems [10, 24, 25, 27, 48, 50, 60, 66, 69, 77, 78]. For instance, the authors in [27] proposed a pre-training and fine-tuning method to reduce the gender bias in career recommendations without sacrificing the performance of the learnt system. Yao et al [69] proposed a regularized-based approach to improve the fairness of recommender systems by incorporating an unfairness term into the objective function. The authors in [56] propose a re-ranking algorithm to post-process the recommendation list to achieve fairness. However, these existing works are “intrusive” in the sense that they need to modify either the original training data or the learning algorithms of the recommender systems, both of which are quite cumbersome in practice. These limitations motivate us to pursue a different approach in this paper.
Our work: In this paper, we propose a new approach named FairRoad (fair recommendation with optimized antidote data) to achieve fairness for recommender systems without modifying the original data nor recommendation algorithms. The basic idea of our proposed FairRoad method is that we assume the service provider (e.g., the companies who provide recommendations to users) is able to inject $m$ antidote users into the recommender systems. Each antidote user can at most give rating scores to $n$ items (called filler items). Clearly, by injecting an appropriate amount of antidote users with carefully crafted ratings for filler items, the learning outcome of the recommender systems will be changed, so that fairness risks of the recommender systems can be mitigated. Toward this end, a key question naturally arises: given some targeted unfairness metric (more on this later), how to optimally select filler items for each antidote user and assign rating scores to those items, such that the unfairness score is minimized after training the model on the original and injected antidote data?

To answer this fundamental question, we formulate our antidote data generation problem as a mathematical optimization problem, for which the objective function is to minimize some targeted unfairness metric. However, it turns out that the formulated optimization problem for antidote data generation is a highly challenging integer programming problem, which is computationally intractable. To address this challenge, we propose several effective techniques to lower the complexity of the formulated optimization problem. First, we convert the rating scores of integer values to continuous variables, solve the relaxed continuous optimization problem, and then project the solution back to integer rating scores. Second, instead of computing the rating scores of $m$ antidote users all at once, we optimize rating scores of antidote users one by one in a sequential fashion. Specifically, in each iteration, we add one antidote user into the recommender system and compute the optimized rating scores for this antidote user.

A major novelty of this paper is that, upon completing the algorithmic design of antidote data for a single unfairness performance metric, we go one step further to consider optimizing antidote data for optimizing or balancing multiple unfairness metrics. This is motivated by the observations from recent studies [4, 7, 48] that improving one unfairness metric for recommender systems may often worsen the performance of another unfairness metric. That is, different unfairness metrics may conflict with one another. So far, most of the existing works in the literature are focused on how to reduce the fairness risks in terms of a single unfairness metric. However, how to simultaneously reduce the fairness risks of multiple unfairness metrics remain largely unexplored. Note that this multi-unfairness antidote data optimization problem is in essence a multi-objective optimization problem. In the literature of multi-objective optimization, a classical problem formulation is to optimize the weighted sum of multiple unfairness metrics. However, in practice, it is often hard to specify the weight for each unfairness performance metric. Moreover, even if the weights for different unfairness metrics are known, simply using the gradient descent algorithm that follows the gradient directions of the weighted sum of multiple unfairness metrics is often inefficient. This is due to the fact that most of the unfairness metrics of practical interest have highly complex geometric characteristics (convexity, curvature, etc.).

To address the above challenges, in our paper, we propose a gradient-deflecting approach motivated by [62, 71] to improve the performance of multi-unfairness antidote data generation. Specifically, given two unfairness performance metrics that the service provider wants to optimize, we first compute the gradients of the objective functions of these two unfairness metrics. Then, we examine whether gradients are conflicting by computing the cosine similarity between gradients. If the cosine similarity is positive, which indicates the two gradients are not in conflict, we then sum the two gradients and a leverage projected gradient descent approach to generate rating scores of antidote users. If on the other hand the cosine similarity is negative (i.e., gradients are in conflict), we modify each gradient by projecting one gradient onto the normal plane of the other one, which helps avoid being trapped at some point that has high curvature. Finally, the two altered gradients are added to obtain the deflected gradient, which will guide the antidote data generation process.

We evaluate our FairRoad antidote data generation method on four synthetic and one real-world datasets. We show that our FairRoad approach could improve the fairness of recommender systems with a small amount of antidote data. Our FairRoad could reduce 31.25% of overestimation unfairness when only injecting 2% of antidote users into the synthetic dataset; 19.05% of value unfairness is reduced with 2% of antidote users on Yelp dataset. Lastly, we summarize our main contributions in this paper as follows:

- We propose a new method called FairRoad based on antidote data optimization to improve fairness for recommender systems.
- Based on the algorithmic framework of FairRoad for a single unfairness performance metric, we further propose a new approach to improve the performance of multi-unfairness antidote data generation for recommender systems.
- Extensive experiments on four synthetic and one real-world benchmark demonstrate that our proposed FairRoad algorithms are effective in mitigating the unfairness in recommender systems and outperform existing baselines.

2 PRELIMINARIES AND RELATED WORK

In this section, we first provide a quick overview of the basics of recommender systems and their fairness metrics in Section 2.1 and Section 2.2 to familiarize readers with the necessary background. Then, in Section 2.3, we introduce some related work on fairness, thus putting our work in a comparative perspective. Table 1 summarizes the key notation used in this paper.

2.1 Basics of Recommender Systems

In this paper, we focus on recommender systems based on collaborative filtering using matrix factorization [37], which is one of the most popular approaches for implementing recommender systems. Toward this end, we use $\mathcal{U}$ and $\mathcal{I}$ to represent the set of users and items, respectively. We let $|\mathcal{U}|$ and $|\mathcal{I}|$ represent the numbers of users and items, respectively. Each user is associated with some identity (e.g., male or female). Let $R \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ denote the observed rating matrix, and $\Omega$ be all observable entries in $R$. An entry $r_{ui}$ in matrix $R$ means the rating score that user $u$ gives to item $i$, where $u \in \mathcal{U}$, $i \in \mathcal{I}$. We further let $p_u \in \mathbb{R}^{d}$ and $q_i \in \mathbb{R}^{d}$ be
Table 1: Notation.

| Notation | Definition |
|----------|------------|
| U        | Set of users |
| I        | Set of items |
| |U| | Number of users |
| |I| | Number of items |
| D        | Set of disadvantaged users |
| A        | Set of advantaged users |
| Uᵢ       | Set of users who rate item i |
| Iᵢ       | Set of items rated by user u |
| Dᵢ       | Set of disadvantaged users who rate item i |
| Aᵢ       | Set of advantaged users who rate item i |

the d-dimensional latent vectors of user u and item i, respectively. In matrix-factorization-based recommender systems, the objective function can be defined as follows:

$$\arg\min_{P,Q} \sum_{(u,i) \in \Omega} (r_{ui} - p_u^T q_i)^2 + \lambda \left( \|P\|_F^2 + \|Q\|_F^2 \right), \quad (1)$$

where $P = [p_1, \ldots, p_U]$, $Q = [q_1, \ldots, q_I]$, $\|.\|_F$ is the Frobenius norm, and $\lambda$ is the regularization term. The predicted rating of user $u$ gives to item $i$ can be computed as $\hat{r}_{ui} = p_u^T q_i$, where $p_u^T$ is the transpose of vector $p_u$.

2.2 Fairness Metrics

To date, there have been several fairness metrics for recommender systems proposed in the machine learning community, including individual fairness [18], group fairness [12], equalized odds [26] and demographic parity [11]. In this paper, we consider the four unfairness metrics that are applicable for matrix-factorization-based recommender systems for user groups that can be categorized into advantaged and disadvantaged users (e.g., the advantaged and disadvantaged groups may be male and female users, respectively).

1) Value Unfairness [69]: The value unfairness $M_{val}$ metric measures the discrepancy in signed prediction errors between advantaged and disadvantaged users. Value unfairness emerges when users in one group (advantaged or disadvantaged) always receive higher or lower predictions than their actual preferences. Specifically, the value unfairness $M_{val}$ can be computed as follows:

$$M_{val} = \frac{1}{|I|} \sum_{i \in I} |(E_D[r]_i - E_A[r]_i) - (E_A[r]_i - E_A[r]_i)|, \quad (2)$$

where $D$ and $A$ denote the sets of disadvantaged and advantaged users, respectively; $E_D[r]_i$ is the average predicted rating score for item $i$ from disadvantaged users, $E_A[r]_i$ is the average predicted rating score for item $i$ from advantaged users, and $E_D[r]_i$ and $E_A[r]_i$ are the average rating scores for item $i$ from the disadvantaged and advantaged users, respectively. More specifically, $E_D[r]_i$ can be calculated as:

$$E_D[r]_i = \frac{1}{|D_i|} \sum_{u \in D_i} \hat{r}_{ui}, \quad (3)$$

where $D_i$ is the set of disadvantaged users who rate the item $i$.

2) Absolute Unfairness [69]: The absolute unfairness $M_{abs}$ metric captures the discrepancy in absolute prediction errors between advantaged and disadvantaged users. The absolute unfairness $M_{abs}$ is large when users in one group have small estimation errors and users in the other group have large estimation errors. The absolute unfairness $M_{abs}$ can be computed as:

$$M_{abs} = \frac{1}{|I|} \sum_{i \in I} |E_D[r]_i - E_A[r]_i| - |E_A[r]_i - E_A[r]_i|, \quad (4)$$

3) Overestimation Unfairness [69]: Overestimation unfairness $M_{over}$ measures the discrepancy in how much the predictions overestimate the true ratings. $M_{over}$ is large when the systems constantly predict higher rating scores than the true rating scores. Overestimation unfairness $M_{over}$ is computed as the following:

$$M_{over} = \frac{1}{|I|} \sum_{i \in I} \max \{0, E_D[r]_i - E_A[r]_i\} - \min \{0, E_A[r]_i - E_A[r]_i\}. \quad (5)$$

4) Non-parity Unfairness [69]: Non-parity unfairness $M_{par}$ captures the absolute unfairness of predictions for advantaged and disadvantaged users. Non-parity unfairness $M_{par}$ is computed as the absolute difference between the average predicted rating scores of advantaged and disadvantaged users. To be specific, $M_{par}$ can be computed as:

$$M_{par} = |E_D[r]_i - E_A[r]_i|, \quad (6)$$

where $E_D[r]_i$ and $E_A[r]_i$ are the average predicted rating scores of disadvantaged and advantaged users, respectively.

2.3 Related Work

Noting that recommender systems are a quintessential application of machine learning technologies, in this section, we provide a quick overview on the broader algorithmic fairness research for machine learning systems. In recent years, algorithmic fairness research has received increasing attention in the learning community [7–10, 15, 23–27, 31, 33–35, 39–42, 45, 46, 48–50, 53, 55, 59, 65, 66, 68, 72, 73, 76–78]. Among these, one major line of research is focused on fairness in classification and regression, with a variety of fairness notions being proposed [5, 11, 12, 18, 26, 58, 69, 72]. For example, individual fairness [18] requires that two similar individuals should receive similar algorithmic outcomes. Equal odds [26] requires that the true positive and false positive rates should be equivalent for different groups. Disparate mistreatment [72] ensures that classification error rate should be the same among different groups. Inspired by the fairness notions from the machine learning community, Yao et. al [69] propose several unfairness metrics that can be used to matrix-factorization-based recommender systems, including value unfairness, absolute unfairness, overestimate unfairness and non-parity unfairness. These unfairness metrics all measure the difference in estimation error of rating scores for disadvantaged and advantaged users.

Recently, there has also been another line of work focusing on mitigating the bias in machine learning and recommender systems. According to different strategies that are used to improve fairness, these approaches fall into three categories: pre-processing,
We assume that the service provider knows all details of the targeted recommender systems, including all the rating data, the parameters and algorithms used in recommendation model. We note that, although appearing to be a very strong assumption, this setting is reasonable in practice since the rating data could be public and the service provider itself knows the details of the deployed recommender systems.

3.2 Formulating Antidote Data Generation Model as an Optimization Problem

In this subsection, we formally formulate the antidote data generation model as an optimization problem to reduce the unfairness of the targeted recommender systems. Suppose the rating scores of targeted recommender systems are restricted to be integers and can be chosen from the set \( \{r_{\text{min}}, \ldots, r_{\text{max}}\} \), where \( r_{\text{min}} \) and \( r_{\text{max}} \) are the minimum and maximum rating scores, respectively. We assume that the service provider could inject \( \alpha \in [0,1) \) fraction of antidote users into the targeted recommender systems. We let \( \mathcal{U} \) and \( \mathcal{\bar{U}} \) denote the set of original and antidote users, respectively. Therefore, one has \( |\mathcal{\bar{U}}| = \lfloor \alpha |\mathcal{U}| \rfloor \). Each antidote user will rate at most \( n \) items, which are called filler items. For an antidote user \( z \in \mathcal{\bar{U}} \), we let \( r_z \) be the rating score vector of user \( z \) and \( I_z \) be the set of items rated by \( z \). Then, we have \( |I_z| \leq n \). Let \( r_{zi} \) be the rating score that antidote user \( z \) gives to item \( i \), thus one has \( r_{zi} = r_{zi}, i \in I_z \). Let \( M_* \) be a given unfairness metric (e.g., value unfairness, absolute unfairness, overestimation unfairness, or non-parity unfairness). The service provider aims to craft rating scores for each antidote user such that when the recommendation model is trained jointly on the original and antidote rating data, the unfairness metric \( M_* \) is minimized. Putting all modeling together, the unfairness metric minimization problem (UMM) can be formulated as the following:

\[
\text{UMM: min } M_* \quad \text{s.t. } r_{zi} \in \{r_{\text{min}}, \ldots, r_{\text{max}}\}, \quad \forall z \in \mathcal{\bar{U}}, \forall i \in I_z, \quad |I_z| \leq n, \quad \forall z \in \mathcal{\bar{U}}, \quad |\mathcal{\bar{U}}| = \lfloor \alpha |\mathcal{U}| \rfloor.
\]

Note that the above formulation in Problem UMM is a general framework and can be applied to any recommender systems (matrix-factorization-based, association-rule-based, or graph-based) and unfairness metrics. Given a targeted recommender system, solving the optimization problem for a given unfairness metric leads to a specific antidote data generation model for this particular unfairness metric. However, we note that Problem UMM is an integer programming problem and is computationally intractable (the objective function involves the vector \( r_z \) and each value \( r_{zi} \in \mathcal{\bar{U}} \) is an integer variable).

4 OUR SOLUTION

In this section, we propose several efficient and practical techniques to solve the Problem UMM. First, instead of optimizing the rating scores of all antidote users at the same time, we add one antidote user at a time to the targeted recommender system and optimize its rating scores in a sequential fashion. However, even with sequential antidote user injection, it remains challenging to find the optimal rating scores for each antidote user since the rating score...
is an integer variable. Inspired by [20], we relax the rating score to continuous variable, i.e., we let \( x_2 = [x_{2i}, i \in I_2] \) be the relaxed rating score vector of antidote user \( z \), where \( x_{2i} \) is the continuous rating that injected antidote user \( z \) gives to item \( i \) and satisfies \( x_{2i} \in [r_{\min}, r_{\max}] \). We project the continuous rating score back to the integer values in the set \( \{r_{\min}, \ldots, r_{\max}\} \) after solving the optimization problem. Based on the above relaxation, we reformulate Problem UMM as the following optimization problem for antidote user \( z \):

\[
\min_{x_2} \mathcal{L}(x_2) = M_x,
\]

s.t. \( x_{2i} \in [r_{\min}, r_{\max}] \).

In what follows, we will discuss in detail each component of our proposed solution approach.

### 4.1 FairRoad: Antidote Data Generation for Single-Unfairness Optimization

In this subsection, we will show how to solve the optimization problem in (11). For illustrative purpose, we instantiate \( M_x \) with the value unfairness metric \( M_{\text{val}} \) introduced in Section 2.2. Then, the gradient of loss formulated in (11) can be computed as:

\[
g_{\text{val}}(x_2) = \frac{1}{|I_2|} \sum_{i \in I_2} \nabla x_{2i} \left[ (E_D[i] - E_D[r_i]) - (E_A[i] - E_A[r_i]) \right].
\]

The term \( \sum_{i \in I_2} \nabla x_{2i} \left[ (E_D[i] - E_D[r_i]) - (E_A[i] - E_A[r_i]) \right] \) in (12) can further be computed as:

\[
\sum_{i \in I_2} \nabla x_{2i} \left[ (E_D[i] - E_D[r_i]) - (E_A[i] - E_A[r_i]) \right] = \sum_{i \in I_2} \text{sign}((E_D[i] - E_D[r_i]) - (E_A[i] - E_A[r_i])) \cdot \nabla x_{2i} (E_D[i] - E_D[r_i]) - \nabla x_{2i} (E_A[i] - E_A[r_i]).
\]

Following [20, 38], the optimal solution of Problem (15) satisfies the following conditions:

\[
\lambda p_u = \sum_{i \in I_u} (r_{ui} - p_u^T q_i) q_i,
\]

\[
\lambda q_i = \sum_{u \in U_i} (r_{ui} - p_u^T q_i) p_u + (x_{2i} - k^T q_i)k,
\]

where \( I_u \) is the set of items rated by user \( u \), and \( U_i \) is the set of users who rate the item \( i \). We further set the derivatives of (16)–(17) with respect to \( x_{2i} \) to zero. Based on the fact \( (p_u^T q_i) p_u = (p_u p_u^T) q_i \), we have \( \frac{\partial p_u}{\partial x_{2i}} = 0 \) and:

\[
\frac{\partial q_i}{\partial x_{2i}} = \left( \sum_{u \in U_i} p_u p_u^T + kk^T + \lambda I \right)^{-1} k,
\]

where \( I \) is the identity matrix. Further, we can obtain \( H_{x_u}(p_u) \) and \( H_{x_u}(q_i) \) by computing all items \( i \in I \) to finally obtain \( g_{\text{val}}(x_2) \). Also, we note that the gradients of other unfairness metrics (e.g., absolute, overestimation and non-parity unfairness metrics) can also be computed in a similar procedure. To avoid repetition, the details about the derivation of gradients of other unfairness metrics are relegated to Appendix A.

After obtaining the gradient of loss function, we then use the projected gradient descent approach to determine the rating score vector \( x_2 \) for antidote user \( z \). With \( x_2 \), we select \( n \) items with the largest absolute ratings as filler items and truncate ratings to the set \( \{r_{\min}, \ldots, r_{\max}\} \) rounding rating score to the nearest integer value in the set \( \{r_{\min}, \ldots, r_{\max}\} \).

### 4.2 FairRoad: Antidote Data Generation for Multi-Unfairness Optimization

As mentioned in Section 1, one may often prefer to simultaneously achieve good fairness performance in terms of different unfairness
metrics. In this section, we will show how to reduce the bias of multiple unfairness metrics. For simplicity, suppose $M_1$ and $M_2$ are two unfairness metrics that the service provider wants to reduce (the same idea can be generalized to more than two fairness objectives). A natural way to optimize $M_1$ and $M_2$ multi-objective optimization is to solve the following weighted-sum optimization problem:

$$
\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}) = a \cdot M_1 + b \cdot M_2
$$

subject to $\mathbf{x} \in [r_{\text{min}}, r_{\text{max}}].$ (19)

where $a > 0$ and $b > 0$ are some fixed weight parameters that signify the relative importance/preference of the sub-objectives in performing multi-objective optimization. However, solving the problem in Eq. (19) remains challenging in practice. Although it appears natural that Problem (19) can be solved by simply using gradient-descent-type approach, doing so is often inefficient. This is due to the fact that many unfairness-related objective functions (cf. Eqs. (2)-(6)) are often non-smooth and could have large local curvature. As shown in [62, 71], for multi-objective optimization problems with conflicting sub-objective gradients (i.e., the cosine similarity between sub-objective gradients is negative), the gradient descent search process may easily stall and get stuck at “local valley.” To address this challenges, in this paper, we will propose a new approach inspired by [62, 71] to effectively deflect the sub-objective gradients to reduce the unfairness for multiple unfairness metrics.

Specifically, we let $g_1$ and $g_2$ be the gradients of the unfairness sub-objective metrics $M_1$ and $M_2$ in some training iteration, respectively. We first compute the cosine similarity between $g_1$ and $g_2$. If the cosine similarity is positive, as shown in Fig. 1(a), then gradients $g_1$ and $g_2$ are not in conflict, which implies that moving along the combined direction of $g_1$ and $g_2$ is unlikely being trapped, particularly at points with a large local curvature. In this case, one can simply use the gradient of the weighted-sum objective function:

$$
\hat{g} = ag_1 + bg_2. \quad (20)
$$

On the other hand, if the cosine similarity is negative, i.e., $g_1$ and $g_2$ are pointing toward opposite directions over the local manifold of the combined objective function. To avoid being trapped at points with large local curvature, we alter $g_1$ by projecting $g_1$ onto the normal plane of $g_2$, see the green dashed line in Fig. 1(b). The projected gradient $\hat{g}_1$ can be computed as:

$$
\hat{g}_1 = g_1 - \frac{(g_1, g_2)}{\|g_2\|^2}g_2. \quad (21)
$$

Similarly, we replace $g_2$ by its projection onto the normal plane of $g_1$, see the red dashed line in Fig. 1(b). Similarly, the projected gradient $\hat{g}_2$ can be computed as:

$$
\hat{g}_2 = g_2 - \frac{(g_1, g_2)}{\|g_1\|^2}g_1. \quad (22)
$$

The final combined gradient can be calculated as the weighted sum of two altered gradients:

$$
\hat{g} = a\hat{g}_1 + b\hat{g}_2 = ag_1 + bg_2 - a\frac{(g_1, g_2)}{\|g_2\|^2}g_2 - b\frac{(g_1, g_2)}{\|g_1\|^2}g_1. \quad (23)
$$

To summarize, given two unfairness metrics that the service provider wants to minimize, in each training step, we first compute the gradients of the sub-objectives corresponding to the two unfairness metrics, and then compute the final gradient $\hat{g}$ according to Eq. (20) or Eq. (23). With $\hat{g}$, we can again use the projected gradient approach to update rating score for each antidote user. Note that our proposed FairRoad framework could be easily extended to multi-unfairness metrics (more than two unfairness metrics). The way of how to deflect more than two gradients could be found in [62, 71], and we omit it here for briefly.

## 5 EXPERIMENTS

### 5.1 Experimental Settings

1) Datasets: In our paper, we use synthetic and the Yelp datasets to show the effectiveness of our FairRoad approach. For the synthetic data, we construct two user groups (Male, Female) and two item groups (STEM, Non-STEM), where STEM represents the science, technology, engineering, and mathematics courses, and Non-STEM denotes the Non-STEM courses. Inspired by [17, 69], we consider the population imbalance and observation bias when generating the synthetic dataset. Population imbalance occurs when different types of users have different preferences to give high ratings to different types of items. For instance, male students may be more likely to give high ratings to STEM courses compared with female students. Observation bias happens when different types of users have different tendencies to rate different types of items, e.g., male students rate STEM courses with higher frequency relative to females.

We generate the synthetic dataset with binary ratings, i.e., $r_{ui} \in \{-1, 1\}$, where $-1$ and $1$ denote user $u$ dislikes and likes item $i$, respectively. To simulate the population imbalance, we generate a rating matrix according to the rating block model as shown in Table 2(a). In Table 2(a), each value is a probability that a user in a user group likes an item in an item group. For instance, a male student likes a STEM course with probability $\alpha_1$ and dislikes a STEM course with probability $1-\alpha_1$. We use a observation block model to simulate the observation bias, as shown in Table 2(b), where each value is a probability that a user in a user group gives rating score $-1$ or $1$ to an item in this item group. For instance, a male student rates a STEM course with probability $1-\beta_1$. To simplify the simulation, we let $\alpha_1 = \alpha_3 = \alpha_5$, $\beta_1 = \beta_3$. Thus, we can use $\alpha_2$ parameters to control the population imbalance, and $\beta_1$ and

| Table 2: Rating and observation block models for synthetic dataset. |
|-----------------|-----------------|-----------------|
| (a) Rating block model. | (b) Observation block model. |
| **Male** | **STEM** | **Non-STEM** | **Female** | **STEM** | **Non-STEM** |
| $\alpha_1$ | $\alpha_2$ | $\alpha_4$ | $\beta_1$ | $\beta_2$ | $\beta_4$ |
### 3) Unfairness Metrics: In our experiments, we use value unfairness, absolute unfairness, overestimation unfairness and non-parity unfairness as our unfairness metrics. In order to measure the effectiveness of our FairRoad method, we use the unfairness score as our evaluation metric. The unfairness score is computed as Eq. (2), Eq. (4), Eq. (5) or Eq. (6) according to different unfairness metrics. The smaller the score, the fairer the recommender systems.

### 4) Parameter Setting: By default, we use the following default parameters: the size of antidote users is 2% of the original users. The number of filler items is \( n = 200 \). We set \( d = 8, \lambda = 0.1, a = 1, b = 1 \).

## 5.2 Experimental Results

### 1) Impact of antidote users size: Table 3 shows the unfairness scores of different unfairness metrics under different approaches. The synthetic dataset (default synthetic dataset) in Table 3 exhibits no population imbalance \((\alpha_1 = \alpha_2 = 0.4)\), and has observation bias \((\beta_1 = 0.2, \beta_2 = 0.1)\). In Table 3, "None" means the unfairness score computed with the original rating data and does not incorporate the unfairness metric as a regularization term. Additional experimental results for different settings of synthetic datasets are relegated to Tables 7-9 in Appendix B. Table 7 summarizes the results when the synthetic dataset exhibits no population imbalance but different observation bias. Table 8 shows the results when the synthetic dataset exhibits different population imbalance and has no observation bias. Table 9 illustrates the results when the synthetic dataset exhibits population imbalance and also has observation bias.

From Table 3, we observe that our proposed FairRoad approach effectively reduces the unfairness of recommender systems with only a small amount of antidote data. For instance, FairRoad reduces 31.25% (the unfairness score decreases from 0.144 to 0.099) of overestimation unfairness when injecting only 2% of antidote users into the synthetic dataset. Also, a 19.05% reduction of value unfairness is achieved with only 2% of antidote users on Yelp dataset (the unfairness score decreases from 0.126 to 0.102).

### Effectiveness of our FairRoad method, we use the unfairness score as our evaluation metric. The unfairness score is computed as Eq. (2), Eq. (4), Eq. (5) or Eq. (6) according to different unfairness metrics. The smaller the score, the fairer the recommender systems.
unfairness. The Regularization method only slightly reduces the unfairness of recommender systems. The approach proposed in Rastegarpanah et al. [50] only achieves sub-optimal performance since it randomly selects \( n \) items as filler items for each antidote user. We also note that FairRoad improves the fairness without sacrificing the performance. For instance, on synthetic data, the Root Mean Square Error (RMSE) without antidote data is 0.630, and the RMSE after injecting 2\% of antidote data generated by FairRoad based on value, absolute, overestimation and non-parity unfairness metrics are 0.633, 0.631, 0.630 and 0.630, respectively. On Yelp dataset, the RMSE without antidote data is 0.697, the RMSE are 0.698, 0.701, 0.697 and 0.698 with 2\% of antidote data generated based on four unfairness notions.

2) Impact of the number of filler items: Figures 2–3 illustrate the results of FairRoad with different number of filler items on the synthetic and Yelp datasets, where the fraction of antidote users is set to 2\%. We do not investigate the impact of number of filler items on Maximum, Minimum and Random data generation approaches since these three methods almost could not reduce the unfairness. For the synthetic dataset, the unfairness score decreases with more filler items for the overestimation and non-parity unfairness metrics. However, the unfairness score does not always decrease with more filler items, implying that there exists an optimal amount of filler items. For instance, on the Yelp dataset, FairRoad performs the best when the number of filler items is 200. The unfairness score then increases as more filler items are used. We also observe that FairRoad outperforms the Regularization and Rastegarpanah et al. [50] methods in most test cases.

3) Antidote users in the representation space: Even though the service provider itself injects some antidote data into the recommender systems to achieve fairness, recent studies have shown that recommender systems are vulnerable to poisoning attack [22, 38], where the attacker could inject some fake users into the systems to degrade the performance of recommender systems. The service provider may arm the recommender systems with anomalies detection capability to detect the fake users and remove the detected fake users from the systems. Based on this observation, the antidote users generated by FairRoad should mimic the rating behavior of original users such that the anomaly detection algorithm deployed by the system will not consider the antidote users as anomalies. In this part, we check whether the antidote users generated by FairRoad can be detected as anomalies in the representation space.
In this experiment, we study the transferability of FairRoad, i.e., generating antidote data based on one unfairness metric could also reduce the unfairness of other unfairness metrics. Here, the fraction of antidote users is set to 2% and the results are shown in Table 4. In Table 4, each row is the source unfairness metric, based on which we generate the antidote data, each column is the targeted unfairness metric. For instance, we generate antidote data according to the value unfairness metric, then compute the unfairness score of absolute unfairness notion after injecting those antidote data into the systems. “Value” in Table 4 means the value unfairness metric. We observe that FairRoad has good transferability in most cases. For instance, on the synthetic dataset, the unfairness score of absolute unfairness metric decreases from 0.107 without antidote data (see Table 3) to 0.099 with 2% of antidote users generated by value unfairness notion.

5) Multiple unfairness metrics optimization: In this experiment, we evaluate the performance of FairRoad in optimizing two unfairness metrics simultaneously. Here, the fraction of antidote users is set to 2% and the results are shown in Table 5, where “Value+Absolute” means FairRoad optimizes value and absolute unfairness metrics simultaneously. We observe that FairRoad reduces the unfairness of multiple unfairness metrics at the same time. For instance, on the synthetic dataset, when FairRoad optimizes value and overestimate unfairness metrics simultaneously. The unfairness score of value unfairness decreases from 0.280 without antidote data (see Table 3) to 0.204 with 2% of antidote users; the unfairness score of overestimate unfairness decreases from 0.144 without antidote data (see Table 3) to 0.104 with only 2% of antidote users.

To see the benefits of deflecting conflicting sub-objective gradients in our FairRoad approach, we also consider the setting where one optimizes two unfairness metrics simultaneously without resolving conflicting gradients, i.e., computes the final gradient only based on Eq. (20). The results of optimizing non-parity and other unfairness metrics simultaneously on the synthetic data are shown in Table 6, where the fraction of antidote users is set to 2%. We observe that, numerically, this approach could increase the unfairness score of targeted unfairness metric. For instance, on the synthetic dataset, the unfairness score of overestimate unfairness decreases from 0.144 without antidote data to 0.104 and with 2% of antidote users. However, the unfairness score of non-parity increases from 0.041 (no antidote data) to 0.052 (2% of antidote users).

6) CONCLUSION

In this paper, we proposed an effective method named FairRoad to achieve fairness for recommender systems. In our FairRoad approach, the key idea is that the service provider can inject a small number of antidote data into the recommender systems to improve the fairness of a targeted recommender system. We formulate the antidote data generation task as an optimization problem, the solution of which provides the ratings for the injected antidote users. We further proposed a gradient-deflecting technique to reduce the unfairness of multiple unfairness metrics simultaneously. Extensive experiments on four synthetic and the real-world Yelp datasets demonstrated that our FairRoad approach significantly improves the fairness of recommender systems with a small amounts of antidote data.

ACKNOWLEDGEMENTS

This work has been supported in part by NSF grants CAREER CNS-2110259, CNS-2112471, CNS-2102233, CCF-2110252, and a Google Faculty Research Award.

REFERENCES

[1] Scott Alfeld, Xiaojin Zhu, and Paul Barford. 2016. Data poisoning attacks against autoregressive models. In Thirtieth AAAI Conference on Artificial Intelligence.

[2] Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. 2019. Discrimination through optimization: How Facebook’s ad delivery can lead to skewed outcomes. arXiv preprint arXiv:1904.02095 (April 2019).

[3] Gender API. [Online]. https://gender-api.com

[4] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. 2021. Fairness in criminal justice risk assessments: The state of the art. Sociological Methods & Research 50, 1 (2021), 3–44.
A DERIVATION OF THE GRADIENTS OF UNFAIRNESS METRICS

A.1 Gradient of absolute unfairness metric

\[
\frac{1}{|I|} \sum_{i \in I} \nabla x_i \left| |E_D[i] - E_D[i]| - |E_A[i] - E_A[i]| \right| = \frac{1}{|I|} \sum_{i \in I} \text{sign} \left( |E_D[i] - E_D[i]| - |E_A[i] - E_A[i]| \right) \cdot \left( \nabla x_i |E_D[i] - E_D[i]| - \nabla x_i |E_A[i] - E_A[i]| \right) \cdot \left( \text{sign}(E_D[i] - E_D[i]) \cdot \frac{1}{|D_i|} \sum_{u \in D_i} \nabla x_{\hat{f}_{ui}} - \text{sign}(E_A[i] - E_A[i]) \cdot \frac{1}{|A_i|} \sum_{u \in A_i} \nabla x_{\hat{f}_{ui}} \right) \tag{24}
\]

A.2 Gradient of overestimation unfairness metric

Let \( C_3 = \max \{0, E_D[i] - E_D[i] \} \) and \( C_4 = \max \{0, E_A[i] - E_A[i] \} \), then we have:

\[
\frac{1}{|I|} \sum_{i \in I} \nabla x_i \left| C_3 - C_4 \right| = \frac{1}{|I|} \sum_{i \in I} \text{sign} \left( C_3 - C_4 \right) \cdot \left( \nabla x_i C_3 - \nabla x_i C_4 \right), \tag{25}
\]

where \( \nabla x_i C_3 \) and \( \nabla x_i C_4 \) can be computed as:

\[
\nabla x_i C_3 = \left\{ \begin{array}{ll}
\frac{1}{|D_i|} \sum_{u \in D_i} \nabla x_{\hat{f}_{ui}}, & E_D[i] - E_D[i] > 0, \\
0, & \text{otherwise}.
\end{array} \right.
\]

\[
\nabla x_i C_4 = \left\{ \begin{array}{ll}
\frac{1}{|A_i|} \sum_{u \in A_i} \nabla x_{\hat{f}_{ui}}, & E_A[i] - E_A[i] > 0, \\
0, & \text{otherwise}.
\end{array} \right.
\]

A.3 Gradient of non-parity unfairness metric

\[
\nabla x_i |E_D[i] - E_A[i]| = \text{sign}(E_D[i] - E_A[i]) \cdot \left( \nabla x_i E_D[i] - \nabla x_i E_A[i] \right) = \text{sign}(E_D[i] - E_A[i]) \cdot \left( \frac{1}{C_5} \sum_{u \in D_i} \sum_{i \in I_u} \nabla x_{\hat{f}_{ui}} - \frac{1}{C_6} \sum_{u \in A_i} \sum_{i \in I_u} \nabla x_{\hat{f}_{ui}} \right), \tag{28}
\]

where \( C_5 \) and \( C_6 \) are the number of ratings provided by disadvantaged and advantaged users, respectively. To be specific, \( C_5 = \sum_{u \in D_i} \sum_{i \in I_u} 1(u, i) \in D_i \), where \( 1(u, i) \in \Omega = 1 \) if \( (u, i) \in \Omega \) and 0 otherwise.

B ADDITIONAL RESULTS FOR SYNTHETIC DATASETS
Table 7: Unfairness scores under different methods on synthetic dataset. The synthetic dataset exhibits no population imbalance ($\alpha_1 = \alpha_2 = 0.4$) and has observation bias ($\beta_1 = 0.2$, $\beta_2 = 0.05$).

| Datasets   | Method                | Value unfairness | Absolute unfairness | Overestimation unfairness | Non-parity unfairness |
|------------|-----------------------|------------------|---------------------|---------------------------|-----------------------|
|            |                       | 0.5% 1% 2% 3%    | 0.5% 1% 2% 3%       | 0.5% 1% 2% 3%             | 0.5% 1% 2% 3%         |
| Synthetic  | None                  | 0.322 0.322 0.322 0.322 | 0.195 0.195 0.195 0.195 | 0.161 0.161 0.161 0.161 | 0.041 0.041 0.041 0.041 |
|            | Regularization        | 0.294 0.294 0.294 0.294 | 0.187 0.187 0.187 0.187 | 0.150 0.150 0.150 0.150 | 0.037 0.037 0.037 0.037 |
|            | Maximum               | 0.322 0.321 0.323 0.316 | 0.189 0.191 0.186 0.194 | 0.159 0.159 0.160 0.156 | 0.042 0.039 0.050 0.046 |
|            | Minimum               | 0.319 0.315 0.318 0.316 | 0.192 0.192 0.190 0.188 | 0.152 0.155 0.151 0.149 | 0.044 0.049 0.058 0.048 |
|            | Random                | 0.313 0.322 0.319 0.314 | 0.189 0.197 0.195 0.189 | 0.160 0.163 0.163 0.154 | 0.039 0.040 0.041 0.044 |
|            | Rastegarpanah et al.  | 0.308 0.302 0.294 0.289 | 0.189 0.185 0.185 0.184 | 0.155 0.154 0.147 0.143 | 0.037 0.035 0.035 0.035 |
|            | FairRoad              | 0.278 0.245 0.215 0.206 | 0.184 0.175 0.173 0.170 | 0.146 0.144 0.123 0.110 | 0.036 0.028 0.027 0.024 |

Table 8: Unfairness scores under different methods on synthetic dataset. The synthetic dataset exhibits population imbalance ($\alpha_1 = 0.4, \alpha_2 = 0.2$) and has no observation bias ($\beta_1 = \beta_2 = 0.2$).

| Datasets   | Method                | Value unfairness | Absolute unfairness | Overestimation unfairness | Non-parity unfairness |
|------------|-----------------------|------------------|---------------------|---------------------------|-----------------------|
|            |                       | 0.5% 1% 2% 3%    | 0.5% 1% 2% 3%       | 0.5% 1% 2% 3%             | 0.5% 1% 2% 3%         |
| Synthetic  | None                  | 0.106 0.106 0.106 0.106 | 0.039 0.039 0.039 0.039 | 0.035 0.035 0.035 0.035 | 0.044 0.044 0.044 0.044 |
|            | Regularization        | 0.093 0.093 0.093 0.093 | 0.025 0.025 0.025 0.025 | 0.044 0.044 0.044 0.044 | 0.041 0.041 0.041 0.041 |
|            | Maximum               | 0.108 0.125 0.109 0.111 | 0.028 0.030 0.041 0.049 | 0.051 0.058 0.048 0.049 | 0.041 0.041 0.042 0.043 |
|            | Minimum               | 0.098 0.096 0.094 0.092 | 0.031 0.029 0.034 0.046 | 0.049 0.048 0.050 0.045 | 0.043 0.042 0.045 0.047 |
|            | Random                | 0.103 0.103 0.103 0.092 | 0.028 0.027 0.026 0.033 | 0.048 0.051 0.052 0.046 | 0.043 0.043 0.044 0.043 |
|            | Rastegarpanah et al.  | 0.105 0.105 0.098 0.092 | 0.028 0.028 0.026 0.026 | 0.050 0.049 0.046 0.043 | 0.041 0.041 0.040 0.040 |
|            | FairRoad              | 0.094 0.086 0.073 0.073 | 0.021 0.021 0.020 0.020 | 0.042 0.040 0.040 0.038 | 0.041 0.039 0.039 0.038 |

Table 9: Unfairness scores under different methods on synthetic dataset. The synthetic dataset exhibits population imbalance ($\alpha_1 = 0.4, \alpha_2 = 0.3$) and has observation bias ($\beta_1 = 0.2, \beta_2 = 0.1$).

| Datasets   | Method                | Value unfairness | Absolute unfairness | Overestimation unfairness | Non-parity unfairness |
|------------|-----------------------|------------------|---------------------|---------------------------|-----------------------|
|            |                       | 0.5% 1% 2% 3%    | 0.5% 1% 2% 3%       | 0.5% 1% 2% 3%             | 0.5% 1% 2% 3%         |
| Synthetic  | None                  | 0.279 0.279 0.279 0.279 | 0.101 0.101 0.101 0.101 | 0.162 0.162 0.162 0.162 | 0.045 0.045 0.045 0.045 |
|            | Regularization        | 0.241 0.241 0.241 0.241 | 0.093 0.093 0.093 0.095 | 0.155 0.155 0.155 0.155 | 0.034 0.034 0.034 0.034 |
|            | Maximum               | 0.272 0.302 0.299 0.263 | 0.099 0.095 0.099 0.102 | 0.157 0.162 0.153 0.129 | 0.042 0.053 0.043 0.052 |
|            | Minimum               | 0.254 0.255 0.246 0.289 | 0.096 0.089 0.097 0.104 | 0.154 0.145 0.143 0.151 | 0.042 0.047 0.052 0.053 |
|            | Random                | 0.278 0.269 0.276 0.267 | 0.101 0.096 0.108 0.099 | 0.164 0.155 0.160 0.156 | 0.053 0.048 0.045 0.046 |
|            | Rastegarpanah et al.  | 0.257 0.256 0.248 0.230 | 0.101 0.099 0.099 0.092 | 0.161 0.154 0.147 0.125 | 0.037 0.034 0.031 0.029 |
|            | FairRoad              | 0.227 0.210 0.157 0.117 | 0.091 0.087 0.078 0.074 | 0.149 0.134 0.127 0.078 | 0.031 0.031 0.027 0.026 |