Fighting Fire with Fire: Using Antidote Data to Improve Polarization and Fairness of Recommender Systems

Bashir Rastegarpanah
Boston University
bashir@bu.edu

Krishna P. Gummadi
MPI-SWS
gummadi@mpi-sws.org

Mark Crovella
Boston University
crovella@bu.edu

ABSTRACT

The increasing role of recommender systems in many aspects of society makes it essential to consider how such systems may impact social good. Various modifications to recommendation algorithms have been proposed to improve their performance for specific socially relevant measures. However, previous proposals are often not easily adapted to different measures, and they generally require the ability to modify either existing system inputs, the system's algorithm, or the system's outputs. As an alternative, in this paper we introduce the idea of improving the social desirability of recommender system outputs by adding more data to the input, an approach we view as providing 'antidote' data to the system. We formalize the antidote data problem, and develop optimization-based solutions. We take as our model system the matrix factorization approach to recommendation, and we propose a set of measures to capture the polarization or fairness of recommendations. We then show how to generate antidote data for each measure, pointing out a number of computational efficiencies, and discuss the impact on overall system accuracy. Our experiments show that a modest budget for antidote data can lead to significant improvements in the polarization or fairness of recommendations.

1 INTRODUCTION

Recommender systems are at the core of many online platforms that influence the choices we make in our daily lives ranging from what news we read (e.g., Facebook, Twitter) and whose products and services we buy (e.g., Amazon, Uber, Netflix) to whom we meet (e.g., OKCupid, Tinder). As users increasingly rely on recommender systems to make life-affecting choices, concerns are being raised about their inadvertent potential for social harm. Recently, studies have shown how recommender systems predicting user preferences

might offer *unfair or unequal* quality of service to indvidual (or groups of) users [5, 8] or lead to *societal polarization* by increasing the divergence between preferences of individual (or groups of) users [13].

Collaborative filtering recommender systems rely on user-provided data to learn models that are used to predict unknown user preferences. As a result, the recommendations made by such systems may carry undesired properties which are inherent in the observed data. A natural approach then is to consider transformations of input data that ameliorate those properties.

In this paper we explore a new approach. Rather than transforming the system's exisiting input data, we investigate whether simply *augmenting the input with additional data* can improve the social desirability of the resulting recommendations. We explore this question by developing a generic framework that can be used to improve a variety of socially relevant properties of recommender systems. Our framework turns a technique that has previously been thought of as anti-social attacks on learning systems into a method with socially desirable outcomes.

As a strategy for improving recommendations, the data augmentation approach has multiple advantages. Adding new input data may be easier than modifying existing data inputs, as when a system is already running. Additional data can be provided to the system by a third-party who does not need the ability to modify the system's existing input, nor the ability to modify the system's algorithms. Further, the approach is applicable to a wide range of socially relevant properties of a system – essentially any property that can be expressed as a differentiable function of the systems inputs (ratings) and/or outputs (predictions).

The framework we develop starts from an existing matrix-factorization recommender system organized according to users and items, that has already been trained with some input (ratings) data. We consider the addition to the system of new users who provide ratings of existing items. The new users' ratings are chosen according to our framework, so as to improve a socially relevant property of the recommendations that are provided to the *original* users. We call the additional ratings provided 'antidote' data (by analogy to existing work studying data poisoning).

In this paper we instantiate the framework by proposing metrics that capture the polarization and unfairness of the system's recommendations. These metrics build on and extend previous proposals, and include measures of both individual and group unfairness. We show how to generate antidote data for these metrics, and we present a number of computational efficiencies that can be exploited. In the process we consider the relationship between improvements to socially-relevant measures and changes to overall system accuracy. Finally, we show that the small amounts of antidote data (typically on the order of 1% new users) can generate a

dramatic improvement (on the order of 50%) in the polarization or the fairness of the system's recommendations.

2 RELATED WORK

In this section, we first discuss how our measures of fairness and polarization for recommender systems relate to those discussed in prior works. Later, we describe how we leverage insights and methods explored in adversarial machine learning to cause social harm towards social good in recommender systems.

Fairness in machine learning and recommender systems: The past years have witnessed a growing awareness about the potential for social harm by the use of machine learning algorithms in life-affecting decision making scenarios [2, 7]. In response, researchers have proposed numerous notions and measures of fairness for machine learning tasks as varied as classification [17, 36, 37, 40], regression [3], ranking [6, 33, 39], and set selection [10]. These proposed notions fall under two broad categoies: those measuring unfairness at the level of *individual users* and those that measure unfairness at the level of *user groups* [14]. The group-level unfairness measures can be further sub-divided into those that prohibit the use of information related to a user's sensitive group membership when making predictions and those that require users belonging to different sensitive groups to receive, on average, *equal quality of service.* The quality of service received by a user can in turn be measured either *conditioned or unconditioned* on the service outcomes deserved by the user.

Compared to learning tasks such as classification and regression, few studies have explored fairness notions in the context of recommender systems. Recently, Burke et al. [8] observed that recommender systems predicting user preferences over items would have to consider fairness from *two-sides* namely, from the perspective of *users* receiving the recommendations and from the perspective of *items* being recommended. Some of the early works by Kamishima et. al. [22–24] focussed on notions of group-level fairness, where the learning model is modified to ensure that item recommendations are independent of users' features revealing sensitive group membership such as race and gender. More recently, Beutel et. al. [5] and Yao et. al. [35] have defined notions of group-level fairness in recommender systems based on the accuracy of predictions across different groupings of users or items.

Novel Contributions: Here, we not only build upon the group-level notions of fariness proposed by Beutel and Yao (by generalizing them to scenarios with more than two groups), but we also extend them to individual-level. We further note that our fairness notions can be applied either from the perspective of users or items.

Mechanims for fair machine learning and recommender systems: Prior works have explored a number of approaches to incorporating fairness in learning models and recommender systems. These approaches can be broadly categorized into those that rely on (i) *pre-processing*, i.e., transform the training data to reduce the potential for unfair outcomes when using traditional learning models [9, 21], (ii) *in-processing*, i.e., change the learning objectives and models to ensure fair outcomes even using unmodified training data [1, 25], and (iii) *post-processing*, i.e., modify potentially unfair outcomes from existing pre-trained learning models [12, 17].

Novel Contributions: In this paper, we explore a different approach to incorporating our fairness notions in recommender systems. Our approach is in contrast to existing approaches to fair recommendations that primarily rely on in-processing [8, 23]. Unlike in-processing approaches, our approach does not require us to modify the recommendation algorithm for each of our desired notions of fairness.

Leveraging adversarial machine learning for social good: Our approach relies on methods that have been traditionally used in adversarial learning literature to cause social harm [20]. Our key insight is that we can retarget adversarial methods designed to "poison" training data and cause social harm to generate "antidote" training data for social good. Specifically, our antidote data generation methods are inspired by prior work on data poisoning attacks on factorization-based collaborative filtering [26].

Most pre-processing approaches target learning new fair (latent and transformed) representations of original data. Recently, Beutel et. al. [4] leveraged adversarial training procedures to remove information about sensitive group membership from the latent representations learned by a neural network. In contrast, our approach leaves the original training data untouched and instead adds new antidote data to achieve fairness objectives. As our evaluation results presented later in the paper will show, by leaving the original training data unmodified, our approach also achieves good overall prediction accuracy (the traditional objective of recommender algorithms).

Polarization: Polarization refers to the degree to which opinions, views, and sentiments diverge within a population. Several prior works have raised and explored concerns that recommender systems might increase societal polarization by tailoring recommendations to individual user's preferences and trapping users in their own "personalized filter bubbles" [15, 31]. Dandekar et. al [13] show how many traditional recommender algorithms used on Internet platforms can lead to polarization of user opinions in society.

Novel Contributions: We propose to measure the polarization of a recommender system as the extent to which predicted ratings for items vary (diverge) across users. Our polarization metric is consistent with those proposed in [13, 27]. We show how our antidote data generation framework can be used to target reducing (or in certain scenarios, increasing) polarization in predicted ratings.

3 OPTIMAL ANTIDOTE DATA PROBLEM

We start by presenting the system setup, notation, and problem definition. Assume $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a partially observed rating matrix of n users and d items such that element x_{ij} denotes the rating given by user i to item j. Let Ω be the set of indices of known ratings in $\mathbf{X}$. Also Ω^i denotes the indices of known item ratings for user i, and Ω_j denotes the indices of known user ratings for item j.

For a matrix $\mathbf{A}$, $P_{\Omega}(\mathbf{A})$ is a matrix whose elements at $(i, j) \in \Omega$ are a_{ij} and zero elsewhere. Similarly, for a vector $\mathbf{a}$, $P_{\Omega_j}(\mathbf{a})$ is a vector whose elements at $i \in \Omega_j$ are the corresponding elements of $\mathbf{a}$ and zero elsewhere. Throughout the paper, we denote the column j of $\mathbf{A}$ by the vector $\mathbf{a}_j$ and the row i of $\mathbf{A}$ by the vector $\mathbf{a}^i$. All vectors are column vectors.

We assume a factorization based collaborative filtering algorithm is applied to estimate the unknown ratings in $\mathbf{X}$, i.e., for each user i
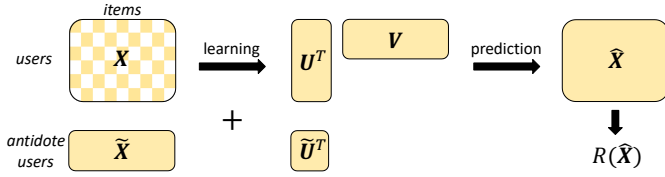
Figure 1: The effect of antidote data on a matrix factorization system. Initially the system learns factors U and V from a partially observed rating matrix X. The latent factors are then used to find the estimated rating matrix $\hat{X}$ which is an input to the socially relevant metric R. Adding antidote ratings $\tilde{X}$ introduces the new user latent factor $\tilde{U}$ and modifies the item latent factor V, generating a new $\hat{X}$ that improves $R(\hat{X})$.

and item j we find ℓ-dimensional representations $\mathbf{u}_i$ and $\mathbf{v}_j$ such that $\ell << min(n, d)$ and the rating x_{ij} is modeled by $x_{ij} \approx \mathbf{u}_i^\mathsf{T} \mathbf{v}_j$.

More specifically, we consider a factorization algorithm Θ that finds factors $\mathbf{U} \in \mathbb{R}^{\ell \times n}$ and $\mathbf{V} \in \mathbb{R}^{\ell \times d}$ by solving the following optimization problem:

$$\underset{\mathbf{U}, \mathbf{V}}{\arg\min} \quad ||P_\Omega(\mathbf{X} - \mathbf{U}^\mathsf{T}\mathbf{V})||_F^2 + \lambda(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2) \qquad (1)$$

where columns of $\mathbf{U}$ are the user latent vectors, and columns of $\mathbf{V}$ are the item latent vectors. The first term in 1 denotes the estimation error over known elements of $\mathbf{X}$ and the second term is an ℓ_2-norm regularizer added to avoid overfitting. The unknown ratings are then estimated by setting $\hat{\mathbf{X}} = \mathbf{U}^\mathsf{T}\mathbf{V}$.

We can think of our factorization algorithm as a function that maps a partially observed rating matrix $\mathbf{X}$ to matrices $\mathbf{U}$ and $\mathbf{V}$, and has additional parameters ℓ and λ, i.e, $\Theta_{\ell,\lambda}(\mathbf{X}) = (\mathbf{U}, \mathbf{V})$. We assume that the factorization rank and the regularizer parameter are set in a validation phase and remain fixed afterwards and we use $\Theta(\mathbf{X})$ and $\Theta_{\ell,\lambda}(\mathbf{X})$ interchangeably throughout the paper.

We use R to denote the socially relevant objective function that we seek to optimize by adding antidote data. R is a function of estimated ratings $\hat{\mathbf{X}}$ and possibly (depending on the objective) other parameters such as original ratings, user labels, etc. For example, consider an objective that minimizes the difference of average estimation errors between two groups of users. In that case, R is a function defined over $\mathbf{X}$, $\hat{\mathbf{X}}$, and another parameter that indicates the group membership of each user. The specific objective functions we study in this paper are presented in Section 5. Now we can formally state the optimal antidote data problem:

PROBLEM 1 (OPTIMAL ANTIDOTE PROBLEM). *Given a partially observed rating matrix* $\mathbf{X} \in \mathbb{R}^{n \times d}$, *a budget* $n' = \alpha n$, *a factorization algorithm* $\Theta_{\ell,\lambda}$, *and an objective function* R, *find the antidote data* $\tilde{\mathbf{X}} \in \mathbb{R}^{n' \times d}$ *such that* R *is optimized when* $\Theta_{\ell,\lambda}$ *is applied jointly on* $\mathbf{X}$ *and* $\tilde{\mathbf{X}}$.

Note that we may want to either maximize or minimize R depending on the objective. Also, although in our notation $\tilde{\mathbf{X}}$ corresponds to a set of artificial users, we can apply problem 1 to generate a set of artificial items by using the symmetry of the problem, i.e., by transposing $\mathbf{X}$.

Although some objective functions have additional parameters such as the original observed ratings ($\mathbf{X}$) or a list of group memberships (which we denote K), adding antidote data only affects

the output of the factorization algorithm and hence the rating estimations $\hat{\mathbf{X}}$. Therefore, we denote the general objective function by $R(\hat{\mathbf{X}})$ instead of $R(\hat{\mathbf{X}}, \mathbf{X}, K)$ for notational convenience. Assuming our goal is to minimize some objective function R, we can rewrite problem 1 as:

$$\underset{\tilde{\mathbf{X}} \in \mathbb{M}}{\arg\min} \quad R(\hat{\mathbf{X}}) \qquad (2)$$

where $\mathbb{M} \subset \mathbb{R}^{n' \times d}$ is the set of feasible antidote data matrices.

Let $\Theta(\mathbf{X}; \tilde{\mathbf{X}})$ denote the factorization algorithm when applied jointly on the original and the antidote data. In this case, the output consists of the item latent vectors forming the columns of factor $\mathbf{V} \in \mathbb{R}^{\ell \times d}$, and the user latent vectors which can be split into a matrix of original users latent vectors $\mathbf{U} \in \mathbb{R}^{\ell \times n}$, and a matrix of antidote users latent vectors $\tilde{\mathbf{U}} \in \mathbb{R}^{\ell \times n'}$; therefore, we have $\Theta(\mathbf{X}; \tilde{\mathbf{X}}) = (\mathbf{U}, \tilde{\mathbf{U}}, \mathbf{V})$.

Furthermore, $\hat{\mathbf{X}}$ is a function of original users latent vectors and item latent vectors[1], i.e., $\hat{\mathbf{X}}(\Theta(\mathbf{X}; \tilde{\mathbf{X}})) = \mathbf{U}^\mathsf{T}\mathbf{V}$. This allows us to write (2) in the explicit form:

$$\underset{\tilde{\mathbf{X}} \in \mathbb{M}}{\arg\min} \quad R(\hat{\mathbf{X}}(\Theta(\mathbf{X}; \tilde{\mathbf{X}}))) \qquad (3)$$

In other words, we are looking for antidote data $\tilde{\mathbf{X}}$ that modifies the outputs of Θ such that $\hat{\mathbf{X}}$ is modified to optimize R. Figure 1 shows a schematic representation of the antidote data effect on matrix factorization models. In the next section, we introduce an iterative method to solve (3).

4 COMPUTING ANTIDOTE DATA

In this section we introduce the framework for generating antidote data. We apply a projected gradient descent/ascent algorithm (**GD/GA**) to optimize the antidote data with respect to a socially relevant objective function. In section 4.1 we review a gradient descent method, introduced in [26], for optimizing data poisoning attacks on matrix factorization models, and which we adapt to optimize antidote data. Then, in section 4.2 we show how the characteristics of the antidote problem can be exploited for significant improvements in algorithmic efficiency.

4.1 A Projected Gradient Descent Approach

In this section we describe a projected gradient descent algorithm to solve the constrained optimization problem (2). A parallel approach is taken in [26] for optimizing data poisoning attacks, which is itself an instance of the more general machine teaching problem introduced in [28]. We note that the framework introduced in [28] can be used to extend the applicability of antidote data approach beyond matrix factorization models.

The algorithm starts from an initial antidote data with size of a given budget. At each iteration, the factorization algorithm is applied jointly on the original data and the current antidote data to find updated factors $\mathbf{U}, \tilde{\mathbf{U}}, \mathbf{V}$, and estimated ratings $\hat{\mathbf{X}}$. Then the gradient of the antidote utility with respect to antidote data at the current point is computed and the algorithm chooses a step size and updates the antidote data. After each update, a projection function is applied to get a feasible solution. In this paper we only

---

[1]Note that here $\mathbf{U}$ and $\mathbf{V}$ are the users and items latent vectors after adding the antidote data to the system, which can be different from initial $\mathbf{U}$ and $\mathbf{V}$.

consider range constraints on the ratings, i.e., for each rating $\tilde{x}_{ij}$ we assume $\mathbb{M}_{min} < \tilde{x}_{ij} < \mathbb{M}_{max}$ where $\mathbb{M}_{min}$ and $\mathbb{M}_{max}$ indicate the minimum and maximum feasible rating in the system. Therefore the projection function simply truncates all the ratings in $\tilde{X}$ at $\mathbb{M}_{min}$ and $\mathbb{M}_{max}$.

Algorithm 1 presents the details of our antidote data optimization method. If the goal is to maximize R, we can apply a gradient ascent algorithm by simply changing the sign of the gradient step in line 6. The learning algorithm $\Theta_{\ell,\lambda}$ is an input to Algorithm 1. This is a realistic assumption in a white-box scenario, i.e., a party with the full knowledge of the recommender system seeks to generate antidote data, which is an important case. However, we emphasize that there are settings in which other parties with only partial knowledge of the system can successfully adopt the antidote data approach as well. First of all, recent work [34] introduces a method for estimating the hyper-parameters of a learning algorithm. Using that method we need not input $\Theta_{\ell,\lambda}$ to Algorithm 1, instead only providing the original factors $(\mathbf{U}, \mathbf{V})$. Moreover, in Section 6 we introduce heuristic algorithms that require less information about the recommender system than does Algorithm 1.

Algorithm 1: Optimizing antidote data via projected gradient descent

Input: Observed ratings $\mathbf{X} \in \mathbb{R}^{n \times d}$, budget n', factorization algorithm $\Theta_{\ell,\lambda}$, utility R, feasible set $\mathbb{M}$

Output: Antidote data $\tilde{X}$

Initialization: initialize $\tilde{X}^{(0)} \in \mathbb{R}^{n' \times d}$, $t = 0$

1 **while** *convergence* **do**
2      $\mathbf{U}, \tilde{\mathbf{U}}, \mathbf{V} = \Theta(\mathbf{X}; \tilde{X}^{(t)})$
3      $\hat{\mathbf{X}} = \mathbf{U}^{\mathsf{T}} \mathbf{V}$
4   Compute $\nabla_{\tilde{X}} R(\hat{\mathbf{X}})$
5   Find step size α
6      $\tilde{X}^{(t+1)} = \tilde{X}^{(t)} - \alpha \nabla_{\tilde{X}} R(\hat{\mathbf{X}})$
7      $\tilde{X}^{(t+1)} = P_{\mathbb{M}}(\tilde{X}^{(t+1)})$
8      $t \leftarrow t + 1$
9 **return** $\tilde{X}^{(t)}$

In order to compute $\nabla_{\tilde{X}} R(\hat{\mathbf{X}})$ in line 4 of algorithm 1, we consider the explicit form of the objective function given in (3). Applying the chain rule we get:

$$\nabla_{\tilde{X}} R(\hat{\mathbf{X}}) = \nabla_{\Theta} R(\hat{\mathbf{X}}) \, \nabla_{\tilde{X}}(\Theta(\mathbf{X}; \tilde{X})) \tag{4}$$

$\nabla_{\tilde{X}}(\Theta(\mathbf{X}; \tilde{X}))$ is the Jacobian matrix that contains partial derivatives of factors $(\mathbf{U}, \tilde{\mathbf{U}}, \mathbf{V})$ with respect to each element in $\tilde{X}$. These partial derivatives can be approximately computed by exploiting the KKT conditions of the factorization problem as explained in [26] [28]. However, in section 4.2 we show cases where the full computation of such partial derivatives is not required and we explain how to derive the necessary elements.

By applying the chain rule one more time on $\nabla_{\Theta} R(\hat{\mathbf{X}})$ we get:

$$\nabla_{\tilde{X}} R(\hat{\mathbf{X}}) = \nabla_{\hat{X}} R(\hat{\mathbf{X}}) \, \nabla_{\Theta} \hat{\mathbf{X}}(\Theta) \, \nabla_{\tilde{X}}(\Theta(\mathbf{X}; \tilde{X})) \tag{5}$$

The first term in (5) is the gradient of the antidote utility with respect to the estimated ratings. In this paper we only consider differentiable utilities, as described in more detail in section 5.

The second term in (5) is the gradient of the estimated ratings with respect to factors $(\mathbf{U}, \tilde{\mathbf{U}}, \mathbf{V})$. This term is straighforward to compute since the ratings are linear in each factor, i.e., $\hat{\mathbf{X}} = \mathbf{U}^{\mathsf{T}} \mathbf{V}$.

In this paper we do not make assumptions (e.g. convexity) about the antidote utility other than being differentiable; the framework is a general method to improve a socially relevant metric rather than one that seeks the global optimum of function R. However, we note that introducing antidote objectives with certain provable properties, which can provide convergence guarantees or more efficient ways to find the step size in Algorithm 1, is a potential direction for future research.

4.2 Efficient Computation of the Gradient Step

In this section we show how to further simplify (5) to make the update step of Algorithm 1 more efficient.

First, we write $\nabla_{\Theta} \hat{\mathbf{X}}(\Theta)$ in terms of the block matrices that contain the partial derivatives of the estimated ratings in $\hat{\mathbf{X}}$ with respect to each factor $\mathbf{U}, \tilde{\mathbf{U}}, \mathbf{V}$, i.e.,[2], $\left[\frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{U}}, \frac{\partial \hat{\mathbf{X}}}{\partial \tilde{\mathbf{U}}}, \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{V}} \right]$. Notice that $\hat{\mathbf{X}} = \mathbf{U}^{\mathsf{T}} \mathbf{V}$ does not depend on $\tilde{\mathbf{U}}$ and therefore $\frac{\partial \hat{\mathbf{X}}}{\partial \tilde{\mathbf{U}}} = \mathbf{0}$.

Furthermore, we write $\nabla_{\tilde{X}}(\Theta(\mathbf{X}; \tilde{X}))$ in terms of the block matrices that contain the partial derivatives of each factor $\mathbf{U}, \tilde{\mathbf{U}}, \mathbf{V}$ with respect to each element in $\tilde{X}$, i.e., $\left[(\frac{\partial \mathbf{U}}{\partial \tilde{X}})^{\mathsf{T}}, (\frac{\partial \tilde{\mathbf{U}}}{\partial \tilde{X}})^{\mathsf{T}}, (\frac{\partial \mathbf{V}}{\partial \tilde{X}})^{\mathsf{T}} \right]^{\mathsf{T}}$. Assuming that an infinitesimal change in $\tilde{x}_{ij}$ only results in first order updates in vectors $\tilde{\mathbf{u}}_i$ and $\mathbf{v}_j$, we get $\frac{\partial \mathbf{U}}{\partial \tilde{X}} = \mathbf{0}$.

Exploiting the fact that $\frac{\partial \mathbf{U}}{\partial \tilde{X}} = \frac{\partial \hat{\mathbf{X}}}{\partial \tilde{\mathbf{U}}} = \mathbf{0}$, we can simplify (5) to:

$$\nabla_{\tilde{X}} R(\hat{\mathbf{X}}) = \nabla_{\hat{X}} R(\hat{\mathbf{X}}) \, \frac{\partial \hat{\mathbf{X}}}{\partial \mathbf{V}} \, \frac{\partial \mathbf{V}}{\partial \tilde{X}} \tag{6}$$

Now we derive $\frac{\partial R(\hat{\mathbf{X}})}{\partial \tilde{x}_{ij}}$ for each element of the antidote data $\tilde{x}_{ij}$. Let $\mathbf{v}_1, \dots, \mathbf{v}_d$ be the item vectors forming the columns of V. Then starting from the last term in (6) and assuming first order updates, we know that $\frac{\partial \mathbf{v}_k}{\partial \tilde{x}_{ij}}$ is non-zero only if $k = j$ and can be approximately computed as[3]:

$$\frac{\partial \mathbf{v}_j}{\partial \tilde{x}_{ij}} = \left(\sum_{i \in \Omega_j} \mathbf{u}_i \mathbf{u}_i^{\mathsf{T}} + \tilde{\mathbf{U}} \tilde{\mathbf{U}}^{\mathsf{T}} + \lambda \mathbf{I}_\ell \right)^{-1} \tilde{\mathbf{u}}_i \tag{7}$$

On the other hand, $\frac{\partial \hat{x}_{lk}}{\partial \mathbf{v}_j} = \mathbf{u}_l^{\mathsf{T}}$ if $k = j$ and an ℓ-dimensional zero vector otherwise. Therefore, we need to compute $\frac{\partial R(\hat{\mathbf{X}})}{\partial \hat{x}_{lk}}$ only for $k = j$ and we have:

$$\frac{\partial R(\hat{\mathbf{X}})}{\partial \tilde{x}_{ij}} = \left(\sum_{l=1}^{n} \frac{\partial R(\hat{\mathbf{X}})}{\partial \hat{x}_{lj}} \frac{\partial \hat{x}_{lj}}{\partial \mathbf{v}_j} \right) \frac{\partial \mathbf{v}_j}{\partial \tilde{x}_{ij}} \tag{8}$$

Let $\mathbf{G}$ be a matrix formed by reshaping $\nabla_{\hat{X}} R(\hat{\mathbf{X}})$ into an $n \times d$ matrix such that $g_{ij} = \frac{\partial R(\hat{\mathbf{X}})}{\partial \hat{x}_{ij}}$. Then we can write (8) as:

$$\frac{\partial R(\hat{\mathbf{X}})}{\partial \tilde{x}_{ij}} = \mathbf{g}_j^{\mathsf{T}} \mathbf{U}^{\mathsf{T}} \mathbf{S}_j^{-1} \tilde{\mathbf{u}}_i \tag{9}$$

---

[2]For matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{r \times s}$, we use $\frac{\partial \mathbf{A}}{\partial \mathbf{B}}$ to denote an $mn \times rs$ matrix that contains the partial derivatives $\frac{\partial a_{ij}}{\partial b_{k\ell}}$ for each a_{ij} and $b_{k\ell}$.

[3]Details are provided in appendix A.1 of the full version of this paper [32].

where $S_j = \sum_{i \in \Omega_j} \mathbf{u}_i \mathbf{u}_i^\mathsf{T} + \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\mathsf{T} + \lambda \mathbf{I}_\ell$.

By using (9) instead of the general formula in (5) we can significantly reduce the number of computations required for finding the gradient of the utility function with respect to the antidote data. Furthermore, the term $\mathbf{g}_j^\mathsf{T} \mathbf{U}^\mathsf{T} S_j^{-1}$ appears in all the partial derivatives that correspond to elements in column j of $\tilde{\mathbf{X}}$ and can be precomputed in each iteration of the algorithm and reused for computing partial derivatives with respect to different antidote users.

5 SOCIAL OBJECTIVE FUNCTIONS

The previous section developed a general framework for improving various properties of recommender systems; in this section we show how to apply that framework specifically to issues of polarization and fairness.

As described in Section 2, polarization is the degree to which opinions, views, and sentiments diverge within a population. Recommender systems can capture this effect through the ratings that they present for items. To formalize this notion, we define polarization in terms of the variability of predicted ratings when compared across users. In fact, we note that both very high variability, and very low variability of ratings may be undesirable. In the case of high variability, users have strongly divergent opinions, leading to conflict. Recent analyses of the YouTube recommendation system have suggested that it can enhance this effect [29, 30]. On the other hand, the convergence of user preferences, i.e., very low variability of ratings given to each item across users, corresponds to increased homogeneity, an undesirable phenomenon that may occur as users interact with a recommender system [11]. As a result, in what follows we consider using antidote data in both ways: to either increase or decrease polarization.

As also described in Section 2, unfairness is a topic of growing interest in machine learning. Following the discussion in that section, we consider a recommender system fair if it provides equal quality of service (i.e., prediction accuracy) to all users or all groups of users [37].

Next we formally define the metrics that specify the objective functions associated with each of the above objectives. Since the gradient of each objective function is used in the optimization algorithm, for reproducibility we provide the details about derivation of the gradients in the full version of this paper (appendix A.2 in [32]).

5.1 Polarization

To capture polarization, we seek to measure the extent to which the user ratings *disagree*. Thus, to measure user polarization we consider the estimated ratings $\hat{\mathbf{X}}$, and we define the polarization metric as the normalized sum of pairwise euclidean distances between estimated user ratings, i.e., between rows of $\hat{\mathbf{X}}$. In particular:

$$R_{pol}(\hat{\mathbf{X}}) = \frac{1}{n^2 d} \sum_{k=1}^{n} \sum_{l>k} ||\hat{\mathbf{x}}^k - \hat{\mathbf{x}}^l||^2 \qquad (10)$$

The normalization term $\frac{1}{n^2 d}$ in (10) makes the polarization metric identical to the following definition: [4]

$$R_{pol}(\hat{\mathbf{X}}) = \frac{1}{d} \sum_{j=1}^{d} \sigma_j^2 \qquad (11)$$

where σ_j^2 is the variance of estimated user ratings for item j. Thus this polarization metric can be interpreted either as the average of the variances of estimated ratings in each item, or equivalently as the average user disagreement over all items.

5.2 Fairness

Individual fairness. For each user i, we define ℓ_i, the loss of user i, as the mean squared estimation error over known ratings of user i:

$$\ell_i = \frac{||P_{\Omega^i}(\hat{\mathbf{x}}^i - \mathbf{x}^i)||_2^2}{|\Omega^i|} \qquad (12)$$

Then we define the individual unfairness as the variance of the user losses:[5]

$$R_{indv}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l>k} (\ell_k - \ell_l)^2 \qquad (13)$$

To improve individual fairness, we seek to minimize R_{indv}.

Group fairness. Let I be the set of all users/items and $G = \{G_1 \ldots, G_g\}$ be a partition of users/items into g groups, i.e., $I = \bigcup_{i \in \{1, \ldots, g\}} G_i$. We define the loss of group i as the mean squared estimation error over all known ratings in group i:

$$L_i = \frac{||P_{\Omega_{G_i}}(\hat{\mathbf{X}} - \mathbf{X})||_2^2}{|\Omega_{G_i}|} \qquad (14)$$

For a given partition G, we define the group unfairness as the variance of all group losses:

$$R_{grp}(\mathbf{X}, \hat{\mathbf{X}}, G) = \frac{1}{g^2} \sum_{k=1}^{g} \sum_{l>k} (L_k - L_l)^2 \qquad (15)$$

Again, to improve group fairness, we seek to minimize R_{grp}.

5.3 Accuracy vs. Social Welfare

Adding antidote data to the system to improve a social utility will also have an effect on the overall prediction accuracy. Previous works have considered social objectives as regularizers or constraints added to the recommender model (eg, [8, 25, 38]), implying a trade-off between the prediction accuracy and a social objective.

However, in the case of the metrics we define here, the relationship is not as simple. Considering polarization, we find that in general, increasing or decreasing polarization will tend to decrease system accuracy. In either case we find that system accuracy only declines slightly in our experiments; we report on the specific values in Section 6. Considering either individual or group unfairness, the situation is more subtle. Note that our unfairness metrics will be exactly zero for a system with zero error (perfect accuracy). As a

---

[4] We can derive it by rewriting (10) as $R_{pol}(\hat{\mathbf{X}}) = \frac{1}{d} \sum_{j=1}^{d} \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l>k} (\hat{x}_{kj} - \hat{x}_{lj})^2$.

[5] Note that for a set of equally likely values $x_1, \ldots, x_n$ the variance can be expressed without referring to the mean as: $\frac{1}{n^2} \sum_i \sum_{j>i} (x_i - x_j)^2$.
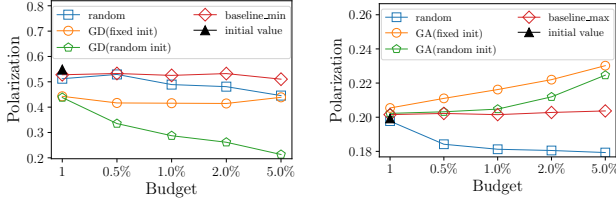
(a) Minimizing polarization   (b) Maximizing polarization

Figure 2: Modifying user polarization.
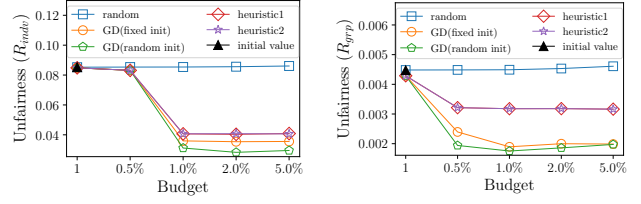
(a) Individual fairness   (b) Group fairness

Figure 3: Improving fairness.

result, it is possible that as the system decreases unfairness, overall accuracy may either increase or decrease. We illustrate these effects in our experiments in Section 6.

6 EFFECTIVENESS

In this section we use the tools developed in previous sections to study the effectiveness of antidote data in varying the polarization and reducing the unfairness of a matrix-factorization based recommender system.

We consider a recommender system that estimates unknown ratings by solving the regularized matrix factorization problem as defined by (1). We implemented an alternating least squares algorithm [16, 19] to find the factors. We use the MovieLens 1M dataset which contains around 1 million ratings of ~4000 movies made by ~6000 users, with ratings on a 5-point scale [18]. We choose the 1000 most frequently rated movies, and use different subsets of users in different experiments as described below.

For each dataset we perform a validation process to choose the hyper-parameters (ℓ, λ) so as to obtain realistic settings. The hyper-parameters are selected based on the average root-mean-square error (RMSE) of the factorization in multiple random splits of observed ratings into training and validation sets. We assume that the hyper-parameters are fixed during the antidote data generation process since the antidote data is generated for a fixed recommender system.

First we show the effectiveness of antidote data in modifying the user polarization as defined in section 5.1. In section 6.2 we describe different heuristics that can significantly speed up the construction of antidote data. Finally, section 6.3 demonstrates the effectiveness of applying antidote data for improving fairness.

6.1 Polarization

To explore modifying user polarization, we choose a random subset of 1000 users yielding a matrix in which 11% of the elements are known. As previously mentioned, it may be of interest to either increase or decrease the polarization metric in different scenarios. We present an example for each case. We do so by taking advantage of the fact that different hyperparameter combinations can yield models that are very close in overall accuracy but that differ significantly with respect to initial user polarization in the system.

In particular, we observe that the average validation RMSE over ten random splits of observed ratings into training and validation sets for $(rank = 8, \lambda = 10)$ is **0.87** and for $(rank = 4, \lambda = 0.1)$ is **0.90**. However, the polarization (R_{pol}) of the estimated rating

matrix for $(rank = 8, \lambda = 10)$ is 0.2 whereas for $(rank = 4, \lambda = 0.1)$ the polarization goes up to 0.55. We use the former setting as an example where the goal is to increase the polarization metric (to avoid homogeneity), and the latter setting as an example of a polarized system where the goal is to reduce polarization.

For each of the maximization and minimization objectives, we compare the performance of the antidote data generation framework with a baseline algorithm. When seeking to minimize polarization, we use `baseline_min`. This algorithm tries to reduce the variance of estimated ratings in each item by setting the ratings given to the corresponding item in the antidote data to the average of known ratings for that item in the original data. When seeking to maximize polization, we use `baseline_max.` This algorithm generates antidote data by setting half of the user ratings in each item to the maximum feasible rating value and the other half of the ratings to the minimum feasible rating value.

Furthermore, we consider two different initializations for the optimization process: in the case of `GD(fixed init)`, all the ratings in the initial antidote data are set to the same value. In the case of `GD(random init)`, we run the optimization multiple times starting from random initializations and return the best solution. Figure 2 compares the effects of adding antidote data constructed by different methods on polarization. After each injection of the antidote data, the new polarization is computed using the original data only, i.e., we ignore the injected data in evaluating polarization. We present our results for different budgets varying from a single antidote user to 5% of the number of original users. We also show the effect of ratings that are randomly generated over the feasible range, when used as antidote data.

Our results show that the antidote data generation framework can successfully either minimize or maximize polarization. Antidote data generated by our method are considerably more effective than the baseline algorithms as well as random data. We observe that a 2% budget is enough to reduce the initial polarization in a polarized setting by 50% and increase the polarization in a less polarized setting by 10%. Furthermore, we observe that random initialization is more effective for minimizing polarization whereas initializing all the antidote ratings from the same value is more effective for maximizing polarization.

To better understand the effect of antidote data on user polarization, in Figure 4 we demonstrate the effect of antidote data with a 1% budget for the minimization case. Note that the effect on estimation error of adding antidote data is negligible: RMSE of rating estimations for known elements changes from **0.80** to **0.83**. In other words, antidote data modifies the prediction model such that its

(a) Effect on per-item polarization

(b) Effect on rating estimations for *Patch Adams (1998)*
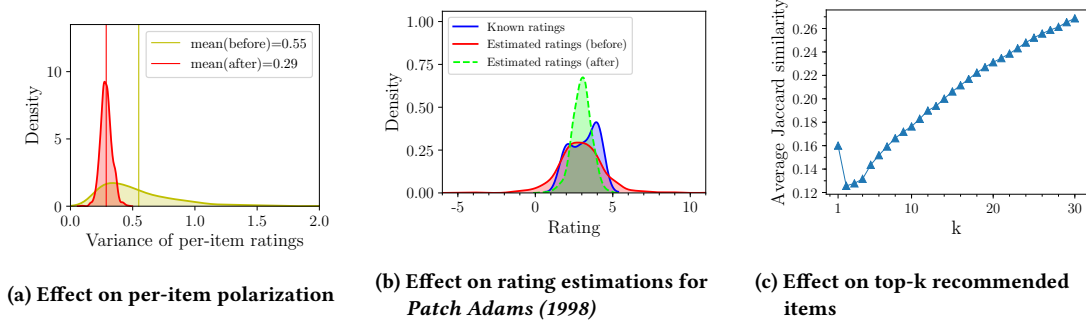
(c) Effect on top-k recommended items

Figure 4: Minimizing polarization with 1% budget.

predictions still approximately agree with the known ratings but the polarization of the new estimated rating matrix is significantly different.

Figure 4a shows the distributions of per-item polarization (σ_j^2 in (11)) along with R_{pol} (the distributional mean) before and after antidote data injection. The figure shows that without antidote data, a small set of items make large contributions to overall polarization – they have quite high variance in ratings, shown by the long distributional tail. The addition of antidote data dramatically reduces this effect, and also significantly lowers R_{pol} from 0.55 to 0.29.

Figure 4b shows the effect of adding antidote data on the estimated ratings of *Patch Adams (1998)*, one of the movies for which the variance of estimated ratings is large before adding antidote data. We observe that the distribution of known ratings for this movie indicates a polarized case with two peaks at 2 and 4. The initial rating estimations in this case lie in an interval that is much larger than the range of observed ratings. Adding antidote data modifies the extreme rating estimations; resulting in a unimodal distribution over the range of original ratings.

While the goal of adding antidote data is to modify the system's predicted ratings, an important use case for such a system is to output the top rated items as the system's recomendations. Hence, it is important to ask how modifying predicted ratings will change the ranking of unrated items, i.e., the output of a top-k recommender system. Therefore, we consider the top-k recommended items on a per-user basis and measure the degree of change in the recommendations before and after adding antidote data. We use the Jaccard similarity of the sets of recommended items to measure this change.

Figure 4c shows the average of Jaccard similarities across all users. Our results show that the antidote data significantly changes the output of a top-k recommender system. For example, adding 1% additional antidote data changes the top-recommended item for 84% of all users. We observe that, in general, as the number of considered top items grows, the effect lessens (Jaccard similarity grows). However, the changes in the set of recommended items are still significant up to $k = 30$.

6.2 Heuristic Algorithms

In this section we introduce heuristic algorithms that dramatically reduce the computational cost of antidote data generation. Notice that the computational cost of Algorithm 1 is dominated by performing the matrix factorization algorithm (evaluating Θ) in each

(a) Minimizing polarization
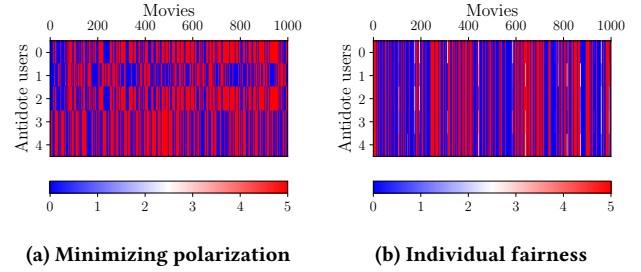
(b) Individual fairness

Figure 5: Optimal antidote data with 0.5% budget.

pass through the gradient descent loop. The heuristics are designed based on various approximations that can be made in different steps of Algorithm 1 to minimize the number of times Θ is evaluated. The approximations are motivated by certain patterns observed in the antidote data generated by Algorithm 1.

Figure 5 shows the antidote data generated by `GD(random init)` for minimizing polarization (Fig. 5a) and minimizing individual unfairness (Fig. 5b). We observe that: (i) most of the ratings in the resulting antidote data are equal to one of the boundary values in the feasible set, $\mathbb{M}_{min}$ or $\mathbb{M}_{max}$ (0 or 5 in our experiments), and (ii) in the fairness case, most of the users (rows) in the antidote data converge to a nearly-identical pattern of ratings over items, even if they are initialized with different random values.

Based on the above observations, in our evaluations we consider two heuristics for generating antidote data for fairness. The first (`heuristic1`) offers considerable computational savings, and the second (`heuristic2`) offers even more savings, while additionally removing the need for access to the factorization algorithm or its hyper-parameters (ℓ, λ)[6].

`heuristic1` reduces the number evaluations of Θ to a single call by combining observations (i) and (ii). It works by considering the addition of only a single row of antidote data, and computes gradients for that row. Rather than performing gradient descent over a series of small steps, it then simply sets each value in the antidote data row to either $\mathbb{M}_{min}$ or $\mathbb{M}_{max}$ depending on the sign of the gradient. It then replicates the resulting row as many times as dictated by the antidote data budget.

In the case of `heuristic2`, in addition to using the above observations, we approximate the direction of the gradient without

---

[6]Pseudocodes are provided in appendix B of the full version of this paper [32].

Table 1: Effect of antidote data on individual unfairness R_{indv} in the held-out ratings. R_{indv} before antidote is 0.1087.

| Algorithm | Budget | | | | |
	1	0.5%	1%	2%	5%
GD(random init)	0.1086	0.1084	0.1054	0.1157	0.1086
GD(fixed init)	0.1086	**0.1083**	0.0929	0.0985	0.0968
heuristic1	0.1086	0.1084	**0.0816**	0.0800	0.0830
heuristic2	0.1086	0.1084	0.0817	0.0818	**0.0811**

Table 2: Effect of antidote data on group unfairness R_{grp} in the held-out ratings. R_{grp} before antidote is 0.0088.

| Algorithm | Budget | | | | |
	1	0.5%	1%	2%	5%
GD(random init)	0.0087	0.0035	0.0041	0.0042	0.0045
GD(fixed init)	0.0087	0.0042	**0.0040**	**0.0040**	**0.0044**
heuristic1	0.0087	0.0055	0.0056	0.0057	0.0058
heuristic2	0.0087	0.0055	0.0056	0.0057	0.0058

the need to perform matrix factorization, given access to factors **U** and **V**. In this case, access to the factorization algorithm or its hyper-parameters is not required. Notice that (9) can be rewritten as $\frac{\partial R(\hat{X})}{\partial \tilde{x}_{1j}} = \mathbf{a}_j^\mathsf{T} \mathbf{b}_j$ where $\mathbf{a}_j = \left[\mathbf{g}_j^\mathsf{T} \mathbf{U}^\mathsf{T}\right]^\mathsf{T}$ and $\mathbf{b}_j = \mathbf{S}_j^{-1} \tilde{\mathbf{u}}_1$. For sufficient level of regularization λ, we can approximate $\mathbf{a}_j^\mathsf{T} \mathbf{b}_j \approx c\mathbf{a}_j^\mathsf{T} \mathbf{1}_\ell$ where c is a constant and $\mathbf{1}_\ell$ is an ℓ-dimensional vector of 1's. This leads to a modification of **heuristic1** in which all the values in column j of the antidote data are set to $\mathbb{M}_{min}$ or $\mathbb{M}_{max}$ depending on the sign of $\mathbf{g}_j^\mathsf{T} \mathbf{U}^\mathsf{T} \mathbf{1}_\ell$.

6.3 Fairness

In this section we show how antidote data as generated by our various algorithms improves fairness. We again use the MovieLens dataset; to study group fairness, we group movies by genre as specified in the dataset. In contrast to the case for polarization, the fairness objective is a function of both the known and predicted user ratings. Hence we choose the 1000 most active users and the 1000 most frequently rated movies. This gives us a rating matrix in which ~36% of the elements are known. For this dataset we run the matrix factorization algorithm with hyper-parameters ($rank = 8, \lambda = 1$).

To verify that adding antidote data improves the fairness of unseen ratings, we hold out 20% of the known ratings per user as a test set. We use the remaining data (training set) to generate antidote data; we then measure the effectiveness of the resulting antidote data in both training and test sets.

We start by assessing the effect of antidote data on fairness in the training data. We show the impact of antidote data on individual unfairness (R_{indv}) in Figure 3a and on group unfairness (R_{grp}) in Figure 3b. The figures compare the effect of antidote data as generated by four different algorithms: Algorithm 1 with two different initializations as described in Section 6.1, and the two heuristic algorithms introduced in Section 6.2.

The results show that all algorithms improve fairness considerably. In fact most of the benefits of antidote data can be obtained by only adding 1% additional users in the individual fairness case and 0.5% additional users in the group fairness case. The figures also show that the much simpler heuristics, in which all rows of the antidote data are identical, are effective: for individual fairness, they provide almost all the benefits of Algorithm 1 while for group fairness they provide around half of the benefits of Algorithm 1.

Tables 1 and 2 show the resulting values of the individual and group unfairness metrics in the *test set* after antidote data addition for different budgets and different algorithms. We observe that the antidote data generated to reduce unfairness in the training data is also effective for reducing unfairness on the the held-out

test data. The optimal value (minimum unfairness) in each table is highlighted. We observe that even in the test set, a 2% budget using **heuristic1** can reduce individual unfairness by over 25% (from 0.1087 to 0.0800), and group unfairness can be lowered by more than 50% (from 0.0088 to 0.0035) using **GD(random init)** and a 0.5% budget.

(a) Individual fairness
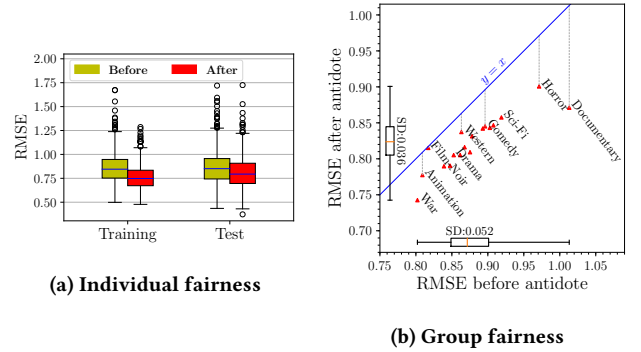
(b) Group fairness

Figure 6: Antidote data effect on fairness.

Figure 6 provides more insight into how adding antidote data reduces individual and group unfairness. In each case, we consider the setting that reaches the minimum unfairness in the test set as presented in tables 1 and 2.

Figure 6a shows the effect of optimal antidote data on per-user RMSEs. The figure demonstrates a number of points. First, adding antidote data results in a model with less variation in per-user RMSE of rating estimations in both training and test sets. Second, a noticeable way in which adding antidote data improves fairness is by reducing the magnitude of the outliers that drive unfairness in both training and testing. Finally, the figure shows that in this example adding antidote data actually improves overall accuracy of the model predictions.

Figure 6b shows the effect of optimal antidote data on per-group RMSE in the test set. For each group (genre) of movies, the corresponding point shows the group's RMSE before and after adding antidote data. Additionally, the boxplots on each axis illustrate the distribution of RMSE values across groups before and after adding antidote data.

First, we observe that all points are below the line $y = x$, i.e., adding antidote data improves the prediction accuracy of all genres and thus the overall accuracy of the model. Moreover, the boxplots show that improvements in rating estimations are so that the cross-group variability in RMSE is decreased to reach a fairer situation.

Finally, we see that outliers particularly benefit from addition of antidote data; this can be seen as larger RMSE improvements in genres that initially had larger RMSE. In particular, *Documentary* and *Horror* have the largest prediction errors before adding antidote data, and their RMSEs are the most improved (furthest below $y = x$) after adding antidote data.

7 CONCLUSION

In this paper we propose a new strategy for improving the socially relevant properties of a recommender system: adding antidote data. We have presented an algorithmic framework for this strategy and applied it to a range of socially important objectives. Using this strategy, one does not need to modify the original system input data or the system's algorithm. We show that the resulting framework can efficiently improve the polarization or fairness properties of a recommender system. We conclude that the developed framework can be a flexible and effective approach to addressing the social impacts of a recommender system.

REFERENCES

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. *arXiv preprint arXiv:1803.02453* (2018).
[2] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Cal. L. Rev.* 104 (2016), 671.
[3] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2017. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409* (2017).
[4] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075* (2017).
[5] Alex Beutel, Ed H Chi, Zhiyuan Cheng, Hubert Pham, and John Anderson. 2017. Beyond globally optimal: Focused learning for improved recommendations. In *Proceedings of the 26th International Conference on World Wide Web.* 203–212.
[6] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. *arXiv preprint arXiv:1805.01788* (2018).
[7] Danah Boyd and Kate Crawford. 2012. Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, communication & society* 15, 5 (2012), 662–679.
[8] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced Neighborhoods for Multi-sided Fairness in Recommendation. In *Conference on Fairness, Accountability and Transparency.* 202–214.
[9] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. 2017. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems.* 3992–4001.
[10] L Elisa Celis, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. 2016. How to be Fair and Diverse? *arXiv preprint arXiv:1610.07183* (2016).
[11] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. 2017. How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility. *arXiv preprint arXiv:1710.11214* (2017).
[12] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 797–806.
[13] Pranav Dandekar, Ashish Goel, and David T. Lee. 2013. Biased assimilation, homophily, and the dynamics of polarization. *Proceedings of the National Academy of Sciences* 110, 15 (2013), 5791–5796.
[14] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference.* ACM, 214–226.
[15] Aniko Hannak, Piotr Sapiezynski, Arash Molavi Kakhki, Balachander Krishnamurthy, David Lazer, Alan Mislove, and Christo Wilson. 2013. Measuring personalization of web search. In *Proceedings of the 22nd international conference on World Wide Web.* ACM, 527–538.
[16] Moritz Hardt. 2014. Understanding alternating minimization for matrix completion. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on.* IEEE, 651–660.
[17] Moritz Hardt, Eric Price, Nati Srebro, et al. 2016. Equality of opportunity in supervised learning. In *Advances in neural information processing systems.* 3315–3323.
[18] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4 (2016), 19.
[19] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. 2015. Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research* 16, 1 (2015), 3367–3402.
[20] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. 2011. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence.* ACM, 43–58.
[21] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.
[22] Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on Recommendation Independence for a Find-Good-Items Task. In *FATREC Workshop on Responsible Recommendation Proceedings.*
[23] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Enhancement of the Neutrality in Recommendation.. In *Decisions@ RecSys.* 8–14.
[24] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2018. Recommendation Independence. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency,* Vol. 81. PMLR, New York, NY, USA, 187–201. http://proceedings.mlr.press/v81/kamishima18a.html
[25] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. 2011. Fairness-aware learning through regularization approach. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on.* IEEE, 643–650.
[26] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in neural information processing systems.* 1885–1893.
[27] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. 2017. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery* 31, 5 (2017), 1480–1505.
[28] Shike Mei and Xiaojin Zhu. 2015. Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners.
[29] Jack Nicas. 2018. How YouTube Drives People to the Internet's Darkest Corners. *Wall Street Journal* (2018). www.wsj.com/articles/how-youtube-drives-viewers
[30] Derek O'Callaghan, Derek Greene, Maura Conway, Joe Carthy, and Pádraig Cunningham. 2015. Down the (White) Rabbit Hole: The Extreme Right and Online Recommender Systems. *Social Science Computer Review* 33, 4 (2015), 459–478.
[31] E. Pariser. 2011. *The Filter Bubble: What the Internet Is Hiding from You.* Penguin Group USA.
[32] Bashir Rastegarpanah, Krishna P. Gummadi, and Mark Crovella. 2018. Fighting Fire with Fire: Using Antidote Data to Improve Polarization and Fairness of Recommender Systems. *arXiv preprint arXiv:1812.01504* (2018).
[33] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. *arXiv preprint arXiv:1802.07281* (2018).
[34] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. *arXiv preprint arXiv:1802.05351* (2018).
[35] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *Advances in Neural Information Processing Systems.* 2925–2934.
[36] Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. 2017. Training fair classifiers. In *AISTATS'17: 20th International Conference on Artificial Intelligence and Statistics.*
[37] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web.* 1171–1180.
[38] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *Artificial Intelligence and Statistics.* 962–970.
[39] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* ACM, 1569–1578.
[40] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *International Conference on Machine Learning.* 325–333.