Pareto Optimality for Fairness-constrained Collaborative Filtering

Qianxiu Hao[1,2], Qianqian Xu[1*], Zhiyong Yang[2], Qingming Huang[1,2,3,4*]
[1]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, China
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
[3]Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, China
[4]Artificial Intelligence Research Center, Peng Cheng Laboratory, Shenzhen, China
qianxiu.hao@vipl.ict.ac.cn,xuqianqian@ict.ac.cn,yangzhiyong@iie.ac.cn,qmhuang@ucas.ac.cn

ABSTRACT

The well-known collaborative filtering (CF) models typically optimize a single objective summed over all historical user-item interactions. Due to inevitable imbalances and biases in real-world data, they may develop a policy that unfairly discriminates against certain subgroups with low sample frequencies. To balance overall recommendation performance and fairness, prevalent solutions apply fairness constraints or regularizations to enforce equality of certain performance across different subgroups. However, simply enforcing equality of performance may lead to large performance degradation of those advantaged subgroups. To address this issue, we formulate a constrained Multi-Objective Optimization (MOO) problem. In contrast to the single objective, we treat the performance of each subgroup equivalently as an objective. This ensures that the imbalanced subgroup sample frequency does not affect the gradient information. We further propose fairness constraints to limit the search space to obtain more balanced solutions. To solve the constrained MOO problem, a gradient-based constrained MOO algorithm is proposed to seek a proper *Pareto optimal* solution for the performance trade-off. Extensive experiments on synthetic and real-world datasets show that our approach could help improve the recommendation accuracy of disadvantaged groups, while not damaging the overall performance.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Collaborative filtering*.

KEYWORDS

Pareto Optimal, Constrained Multi-objective Optimization, Collaborative Filtering

ACM Reference Format:
Qianxiu Hao[1,2], Qianqian Xu[1*], Zhiyong Yang[2], Qingming Huang[1,2,3,4*]. 2021. Pareto Optimality for Fairness-constrained Collaborative Filtering. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475706

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475706

1 INTRODUCTION

With the increasing deployment of recommender systems (RS), the fairness of RS has also attracted much attention in recent years. Fairness in RS is particularly challenging since RS has multiple application fields [40], multiple stakeholders (eg., consumers and suppliers), multiple objectives [23] and complicated sources of biases [41]. In the literature, fair recommendation problem mainly involves three families: fair utilities [36, 40] for each user in a group in the package-to-group recommendation scenario, fair utilities for different stakeholders [23, 27] and fair rankings [3, 5, 28, 37, 38] for different items. Despite there are a large number of fairness-pursued methods, recommendation accuracy disproportions w.r.t consumers are less studied [12, 42]. And we limit ourselves to those scenarios where fair recommendation accuracy for different user groups is desired. Examples of such scenarios include improving the retention of minorities/new users, job recommendation or courses recommendation for different user groups, etc.

Since recommendation models are mainly trained on historical user-item records, they might embody and potentially exacerbate the existing biases in datasets to produce unfair outcomes [33] for certain user groups. In practical scenarios, user-item records are inevitably to have biases. For example, different users have different levels of activity on different social platforms and the demographics or other user characteristics that occur in the dataset are different. To trade-off between overall accuracy and certain fairness metrics across subgroups, prevalent efforts often optimize an overall performance with additional regularizations [3, 41] or constraints [4, 5, 29, 37] based on some fairness metrics, which are typically defined as equality of performance across subgroups. However, this convention is hard to achieve fair/balanced recommendations w.r.t different user groups for at least two causes. One is that the existing biases may cause inherent under-representation of some disadvantaged groups thus lead to undesirable recommendations to those disadvantaged groups. Enforcing equality of performance between those disadvantaged groups and the other advantaged groups may only give a dramatically decreased performance of the advantaged group. The other cause is that there may exist a competitive relationship between different user groups so that the optimal solution for different groups is not consistent. Thus optimizing a single surrogate loss which is the sum of all users might sacrifice the performance of the groups with small sample frequency, thus make unfair training to those users with small sample frequency.

Based on the spirits above, our motivation is to treat each subgroup equivalently as an objective and resort to Multi-Objective Optimization (MOO) scheme to hopefully increase the performance of disadvantaged groups while not damaging the overall performance much. Treating each subgroup equivalently as an objective could not only prevent the gradient information from being dominated by imbalanced subgroups with large sample frequency, but also has the potential to obtain a proper *Pareto optimal* solution [10] across different subgroups. Pareto optimality is a well-known notion in the MOO community and especially focuses on the case where there is no global optimal for multiple objectives. Pareto optimality states that no solution can outperform a Pareto optimal solution over all objectives. In other words, a Pareto optimal solution may perform the best over some objectives and perform worse over other objectives compared with other Pareto optimal solutions. Back to our formulation, since there exist many Pareto optimal solutions and a Pareto optimal may produce extremely unfair/unbalanced performance across subgroups. We intend to seek a proper Pareto optimal solution w.r.t the performance of different user groups, while being as much balanced performance as possible. In this way, we hope to explore the best performances of the disadvantaged groups while not hurting the advantaged groups much.

To produce a more fair/balanced Pareto optimal solution, we further formulate a constrained multi-objective optimization problem to limit the search space of Pareto optimal solutions by fairness constraints. The fairness constraints are built from the losses of a general probabilistic recommendation framework solved by variational autoencoder [19, 25, 44], where we also provide discussions about how some traditional CF methods could be unified into this framework. To solve such a constrained multi-objective optimization problem, we propose a gradient-based algorithm motivated by previous work [9, 24, 35].

The contributions of the paper are summarized as follows:

- We propose to seek a proper *Pareto optimal* solution across multiple user subgroups within certain fairness bound, to hopefully explore the best of disadvantaged groups while not hurting advantaged groups much. Specifically, we formulate a fairness-constrained multi-objective optimization problem and propose a scalable gradient-based optimization algorithm to solve the problem.
- We build a simple fairness constraint from the group losses of a general probabilistic recommendation framework, where we provide insights about how some traditional CF methods could be unified into this framework.
- In experiments, we synthesize user-item interactions with simulated biases to demonstrate the effectiveness of our method. Experimental results on both synthetic data and real-world datasets show that our model could lead to increased performance for the disadvantaged group while not hurting the overall performance much.

2 RELATED WORK

2.1 Fair recommendation

2.1.1 Fairness Notions. Though fair recommendations have recently attracted increasing attention, there is no single definition of what constitutes a fair recommendation since fairness usually depends on the contexts and applications. According to different definitions of fairness in recommendations, existing literature could be divided into mainly three categories: 1) the first category is limited to the package-to-group recommendation scenario [36, 40], where a group of users is recommended with the same package of items. To assure all users within a group are satisfied with the recommended items, fairness is defined as the balance of some predefined fair utilities within group members. 2) Another line of studies considers the satisfaction of different stakeholders (i.e., users and suppliers) [23, 27] in RS, and typically defines fairness as balanced utilities among different stakeholders. 3) The third category study studies fair ranking problems [3–5, 16, 37, 38, 43], where the notion of fair ranking is also not universal.

Despite there are a large number of fairness notions defined for different scenarios, accuracy disproportions remain seldom explored in RS. It is partly because RS usually predicts top-K items that are most likely to be clicked/purchased/liked by a user as positive. Given that the sorting operation is non-differentiable, the ranking-based accuracy metrics (i.e., Recall@k, HitRate@k, etc.) are not directly applicable for trainable fairness. In this paper, we try to seek fair recommendation accuracy among different user groups. To avoid optimizing ranking-based metrics, we define a fairness constraint based on the training losses of different user groups, which stands for the predictive ability.

2.1.2 Fairness-pursued Training Schemes. Besides the fair notions, there are two popular conventions to train a fair recommendation model. One [3, 40, 41] proposes to minimize an overall accuracy-related loss over all users with additional fairness-related regularization terms. The fairness-related regularization could be the approximation [3, 41] of those non-differentiable unfairness metrics to allow for gradient-based optimization. However, as [35] stated that optimizing a weighted linear combination of different losses is valid only when multiple objectives are consistent. Given that overall accuracy and fairness are typically competing with each other, minimizing this surrogate loss might not be an optimal choice. Another convention formulates a constrained optimization problem so as to minimize the overall accuracy-related loss, subjecting to some fairness constraints. The constrained optimization problem is then solved by by post-processing techniques [43], constrained optimization algorithms [4, 5, 29, 37], or reinforcement learning [38], respectively.

Despite the remarkable successes they have made, the above two conventions all optimize an overall accuracy-related loss summed over all users. This scheme may be sensitive to the imbalanced data distribution since the number of user-item records of certain user groups is significantly smaller than that of the other groups. Optimizing an overall loss with additional regularization or constraints may lead the gradient information to highly skew towards the advantaged groups, resulting in the under-representation of those disadvantaged groups.

Seeing this, we proposed to treat the performance of each subgroup equivalently as an objective. This ensures that the imbalanced subgroup sample frequency does not affect the gradient information. We then seek a Pareto optimal among multi-subgroups, trying to explore the best of the disadvantaged groups while not hurting the advantaged groups significantly.

2.2 Multi-objective Optimization

MOO problem aims to optimize multiple objectives simultaneously and make proper trade-offs among them. A most commonly-used strategy is to summarize all objective-specific losses with different weights and optimize a single surrogate loss instead. Given that the preference for each objective is usually unknown, one typically resorts to grid searching or heuristic algorithms [6, 17], which requires exhausting efforts. Moreover, when some objectives are competing with each other, simply minimizing the weighted summary of all losses seems unreasonable. Then, an alternative solution is to find Pareto optimal.

Algorithms to find Pareto optimal have been well-studied in MOO or multi-criteria optimization community, including evolutionary algorithms [8, 45, 46] and gradient-based algorithms [9–11, 30, 31, 35]. The most popular direction is gradient-based methods where the gradients of each task/objective w.r.t shared parameters are used to adaptively determine the updating directions. Of particular relevance to our work is multi-objective gradient-based optimization (MDGA), as developed by [9, 10, 34]. These methods reformulate MOO as a single objective optimization constrained by the other objectives and derive a direction that decreases all objectives by Karush-Kuhn-Tucker (KKT) conditions. This paradigm was later extended to stochastic gradient descent by [30, 31] and recently [24, 35] for effective large-scale optimization.

Our work is based on the efforts [35] that developed an efficient gradient-based algorithm to find a Proper Pareto optimal. The difference is that we further extend it to solve constrained MOO problems.

3 PRELIMINARY

3.1 Objectives

Given a set of users $\mathcal{U} = \{1, 2, \cdots, N\}$, a set of items $\mathcal{I} = \{1, 2, \cdots, M\}$ and their user-item interaction matrix $\boldsymbol{X} \in \{0, 1\}^{N \times M}$, where $\boldsymbol{x}_u = [x_{u,1}, x_{u,2}, \cdots, x_{u,M}]$ is the u-th row of $\boldsymbol{X}$. $x_{u,i} = 1$ means that the implicit feedback between user u and item i is positive (such as like, purchased, or clicked, etc.), $x_{u,i} = 0$ otherwise. Assume there are some attributes that could divide users into T disjoint groups $\mathcal{G}_1, \cdots, \mathcal{G}_T$. Each group i has an objective function $\mathcal{J}_i(\boldsymbol{\theta}), i = 1, 2, \cdots, T$, where $\boldsymbol{\theta}$ is model parameters. Suppose the smaller $\mathcal{J}_i$ is, the better performance the group i has. For fairness consideration, there are C constraints $C_j^{fair}(\boldsymbol{\theta}), j = 1, \cdots, C$.

Different from traditional methods which solve a single objective constrained by fairness criterion or added with fairness regularizations, we aim to solve the following constrained multi-objective optimization problem to obtain a more balanced *Pareto optimal* solution:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \ & \mathcal{J}(\boldsymbol{\theta}) = [\mathcal{J}_1(\boldsymbol{\theta}), \cdots, \mathcal{J}_T(\boldsymbol{\theta})]^\top \\ s.t. \ & C_j^{fair}(\boldsymbol{\theta}) \leq 0, j = 1, \cdots, C. \end{aligned} \tag{1}$$

Pareto optimal solutions are a set of solution enjoying the following properties:

Definition 1 (Pareto optimality[10]). *A solution $\boldsymbol{\theta}$ is called to dominate solution $\boldsymbol{\theta}'$ if $\mathcal{J}_i(\boldsymbol{\theta}) \leq \mathcal{J}_i(\boldsymbol{\theta}')$ for all objective $g \in \{1, 2, \cdots, T\}$ and $\mathcal{J}(\boldsymbol{\theta}) \neq \mathcal{J}(\boldsymbol{\theta}')$. A solution $\boldsymbol{\theta}^\star$ is called Pareto optimal if there exists no solution $\boldsymbol{\theta}'$ that dominates $\boldsymbol{\theta}^\star$.*

Note that there exists a set of Pareto optimal solutions which are called the Pareto set. The image of Pareto optimal solutions is Pareto front. To solve Pareto solutions, for smooth functions, a necessary condition for Pareto optimality is Pareto stationary. Moreover, if the functions involved are convex, Pareto stationarity is sufficient for Pareto optimality. The notion of Pareto stationary is given as:

Definition 2 (Pareto stationarity). *Let $\mathcal{J}_i(\boldsymbol{\theta})(1 \leq i \leq T)$ be smooth objective functions, a solution $\boldsymbol{\theta}^\star$ is called to be Pareto stationary (or Pareto critical) iff $\exists \{w_i\}_{i=1}^T$, $w_i \geq 0$ and $\sum_{i=1}^T w_i = 1$ such that the following conditions hold:*

$$\sum_{i=1}^{T} w_i \nabla \mathcal{J}_i(\boldsymbol{\theta}^\star) = 0 \tag{2}$$

The above Eq. 2 is equivalent to say that at a Pareto optimal point, one cannot find an update direction at point $\boldsymbol{\theta}^\star$, along which every objective-function can be improved. It then leads to a gradient-based update rule, where the update direction $\boldsymbol{d}^\tau$ at the τ-th iteration is determined by first solving the following optimization problem [9]:

$$\begin{aligned} \min_{w_1, \ldots, w_T} \ & \left\| \sum_{i=1}^{T} w_i \nabla \mathcal{J}_i(\boldsymbol{\theta}^\tau) \right\| \\ s.t. \ & \sum_{i=1}^{T} w_i = 1; \ w_i \geq 0 \end{aligned} \tag{3}$$

and then calculate the descent direction as:

$$\boldsymbol{d}^\tau = -\sum_{i=1}^{T} w_i^\tau \nabla \mathcal{J}_i(\boldsymbol{\theta}^\tau). \tag{4}$$

It is proved in [9] that either $\boldsymbol{d}^\tau$ is an update direction that improves all objectives, or the iterations arrive at $\boldsymbol{d}^\tau = \boldsymbol{0}$ and ensure a Pareto stationary solution. [9, 35] have developed efficient algorithm to solve this optimization problem Eq. 3 and this paper extends their algorithms to the constrained MOO problem.

4 METHODOLOGY

In this section, we start by formulating a generative recommendation model in a probabilistic view and obtain the optimization objective by variational autoencoder. We further give some insights that traditional collaborative filtering (CF) methods can be unified into this framework under some simple assumptions. After that, we give the accuracy-related multi-objectives and fairness constraints. Thereafter, we propose an optimization algorithm for the constrained MOO problem.

4.1 A probabilistic recommendation framework

Denote $\boldsymbol{z}_u \in \mathbb{R}^d$ as latent representation of user u, instead of regarding $\boldsymbol{z}_u$ as fixed variables as traditional embedding-based recommendation models do, we let them be *random* variables following a certain distribution $p(z_u)$, i.e., $z_u \sim p(z_u)$. During testing, we predict the items with larger possibility $p(\boldsymbol{x}_u) = \mathbb{E}_{p(z_u)}[p(\boldsymbol{x}_u|z_u)]$ as user u's interested items.

Since we do not know the true distribution $p(z_u)$, naturally, we want to infer a posterior distribution $p(z_u|\boldsymbol{x}_u)$ by Bayes' theorem, i.e., $p(z_u|\boldsymbol{x}_u) = \frac{p_0(z_u)p(\boldsymbol{x}_u|z_u)}{p(\boldsymbol{x}_u)}$, where $p_0(z_u)$ is the prior distribution of the embedding of user u. However, it is intractable due to the intractability of $p(\boldsymbol{x}_u)$. Fortunately, variational inference (see [44] for a detailed review) has developed an equivalent optimization problem by introducing a variational distribution to approximate $p(z_u|\boldsymbol{x}_u)$. To allow for scalable inference, we exploit variational auto-encoder (VAE) [19, 32] to construct the variational distribution as an inference model $q(z_u|\boldsymbol{x}_u)$. It takes as inputs the users' historical interactions $\boldsymbol{x}_u$ and outputs the variational distribution $q(z_u|\boldsymbol{x}_u)$. Naturally, the goal becomes to minimize $KL(q(z_u|\boldsymbol{x}_u)||p(z_u|\boldsymbol{x}_u))$. According to [19], we have

$$\begin{aligned} &KL(q(z_u|\boldsymbol{x}_u)||p(z_u|\boldsymbol{x}_u)) \\ &= -\mathbb{E}_q[\log p(\boldsymbol{x}_u|z_u)] + KL(q(z_u|\boldsymbol{x}_u)||p_0(z_u)) + \log p(\boldsymbol{x}_u) \end{aligned} \tag{5}$$

where q is short for $q(z_u|\boldsymbol{x}_u)$. Equivalently, we have:

$$\log p(\boldsymbol{x}_u) = KL(q(z_u|\boldsymbol{x}_u)||p(z_u|\boldsymbol{x}_u)) + ELBO \tag{6}$$

where

$$ELBO = \mathbb{E}_q[\log p(\boldsymbol{x}_u|z_u)] - KL(q(z_u|\boldsymbol{x}_u)||p_0(z_u)) \tag{7}$$

Since $\log p(\boldsymbol{x}_u)$ (called evidence) is a constant and the KL term is always non-negative, minimizing $KL(q(z_u|\boldsymbol{x}_u)||p(z_u|\boldsymbol{x}_u))$ is equivalent to maximizing the ELBO term (short for Evidence Lower BOund). Then the optimization goal becomes:

$$\max_{\boldsymbol{\theta}} ELBO \tag{8}$$

where $\boldsymbol{\theta}$ is the parameters of the model. The merit of VAE is that it involves only global variational parameters which parameterize the inference model which is shared by all users. This significantly amortizes the cost of training and can directly adapt to new users who have never appeared during training.

4.2 Connection with traditional CF models

Interestingly, the losses of traditional collaborative filtering (CF) [14, 15, 20–22] models can be unified into the above probabilistic framework. The key insight is to hypothesize the cumulative distribution of each user's latent representation $F(z)$ in traditional models as an indicator function $\mathbb{I}(z{=}z_u)$, where $\mathbb{I}(*)$ equals 1 if $*$ holds, and equals 0 otherwise. To avoid notation conflict, here we let z_u be traditional latent representations of user u. Once learned, it is *fixed*. We omit the subscript u and let z be the corresponding *random* variables following the cumulative distribution $F(z)$. Then the expectation of log-likelihood over random variables z can be rewritten as:

$$\begin{aligned} \mathbb{E}_{F(z)}[\log p(\boldsymbol{x}_u|z)] &= \int_z \log p(\boldsymbol{x}_u|z)\mathrm{d}F(z) \\ &= \int_z \log p(\boldsymbol{x}_u|z)\mathrm{d}\mathbb{I}(z{=}z_u) \\ &= \log p(\boldsymbol{x}_u|z_u) \end{aligned} \tag{9}$$

Compared with Eq. 7, the cumulative distribution $F(z)$ of CFs is constrained to an indicator function and is solved by Bayes Theorem (thus the KL term equals to zero). For a typical CF model that aims to learn a score function $s_{\boldsymbol{\phi}}: \mathbb{R}^d \rightarrow \mathbb{R}^M$, mapping z_u to a score vector $s_u = [s_{u,i}]_{i=1}^M$ that represents a user's interests on M items. Herein, $\boldsymbol{\phi}$ is the model parameters to be learned. Assume that the likelihood $p(\boldsymbol{x}_u|z_u)$ is a function of the score vector, i.e., $p(\boldsymbol{x}_u|z_u) = f(\boldsymbol{x}_u, \boldsymbol{s}_u)$ and that the partition function (normalization factor) is independent of $\boldsymbol{\phi}$. Now we show that under different cases, Eq. 9 can induce different loss forms which are typically used in CF models.

Case 1. *Assume $p(\boldsymbol{x}_u|z_u) = \prod_{i=1}^M p(x_{u,i}|z_u)$, if we make $f(\cdot,\cdot)$ an exponential function so that:*

$$p(x_{u,i}|z_u) \propto \exp\left(-\frac{(x_{u,i}-s_{u,i})^2}{2}\right)$$

is a Gaussian distribution, we have:

$$\mathbb{E}_{F(z)}[\log p(\boldsymbol{x}_u|z)] \propto -\sum_{i=1}^M (x_{u,i}-s_{u,i})^2$$

which corresponds exactly to the typical regression loss in recommendation models [15, 20]. For example, if score function s is a linear function, it then becomes the matrix factorization (MF) [20] model.

Case 2. *Again, if we make $f(\cdot,\cdot)$ a sigmoid-like function so that:*

$$p(x_{u,i}|z_u) = p^{x_{u,i}}(1-p)^{1-x_{u,i}}$$

is a binomial distribution $\mathcal{B}(1,p)$, where $p{=}\sigma(s_{u,i}){=}1/(1{+}\exp(-s_{u,i}))$. We can obtain the equivalent form as:

$$\begin{aligned} &\mathbb{E}_{p(z)}[\log p(\boldsymbol{x}_u|z)] \\ &\propto -\sum_{i=1}^M \left[x_{u,i}\log(\sigma(s_{u,i})) + (1-x_{u,i})\log(1-\sigma(s_{u,i}))\right], \end{aligned}$$

which corresponds exactly to the cross-entropy loss used in many recommendation models [14].

In our model, we hypothesize that the user-item interactions $\boldsymbol{x}_u$ are generated from a multinomial distribution, i.e., $p(\boldsymbol{x}_u|z_u) = Multinomial(\boldsymbol{p}_u)$ where $\boldsymbol{p}_u = [p_{u1}, \cdots, p_{uM}]$ and $p_{u,i}$ is the possibility of user u to choose item i, s.t. $\sum_{i=1}^M p_{u,i}{=}1$. Note that $x_{u,i}$ equals 1 or 0, hence:

$$p(\boldsymbol{x}_u|z_u) \propto \prod_{i=1}^M p_{u,i}^{x_{u,i}} = \prod_{i:x_{u,i}=1} p_{u,i} \tag{10}$$

Substituting $p(\boldsymbol{x}_u|z_u)$ into Eq. 8, we have the objective for each user u as:

$$\begin{aligned} \mathcal{J}^{(u)} &= -ELBO \\ &= -\mathbb{E}_q[\log p(\boldsymbol{x}_u|z_u)] + KL(q(z_u|\boldsymbol{x}_u)||p_0(z_u)) \\ &\propto -\sum_{i:x_{u,i}=1} \mathbb{E}_q\left[\log p_{u,i}\right] + KL(q(z_u|\boldsymbol{x}_u)||p_0(z_u)) \end{aligned} \tag{11}$$

Since the above objective contains expectation operation over $p_{u,i}$ w.r.t distribution $q(z_u|\boldsymbol{x}_u)$, the detailed parameterized form of $q(z_u|\boldsymbol{x}_u)$, $p_{u,i}$ and p_0, as well as how to calculate the expectation need to be specified. However, since it is not our focus, we adopt the practices in the recent work [25], which is elaborated in Sec. A in the supplementary materials.

4.3 Multi-objectives

Previous fairness-pursued models typically optimize an overall objective $\frac{1}{N}\sum_{u=1}^{N}\mathcal{J}^{(u)}$ with some additional fairness constraints or fairness regularizations. We proposed to treat the performance of each user group equivalently as an objective and try to seek a proper Pareto optimal solution among these groups. Therefore, we define the performance of user group i as:

$$\mathcal{J}_i(\boldsymbol{\theta}) = \frac{1}{|\mathcal{G}_i|}\sum_{u\in\mathcal{G}_i}\mathcal{J}^{(u)}(\boldsymbol{\theta}). \tag{12}$$

So far, we have given the formulation of our multi-objectives. Next, we proceed to describe our fairness constraints.

4.4 Fairness constraints

Since Pareto optimal solutions are not unique, directly applying Eq. 3, 4 to solve an unconstrained multi-objective optimization only produces a single Pareto optimal solution. In the context of fair recommendation, we might want to have control over the Pareto optimal solutions so that the performance of different groups is not significantly unbalanced.

Seeing the ubiquity of recommendation systems, there is no single definition of what constitutes a fair recommendation, but that fairness depends on the contexts and applications. Let $\mathcal{F}_i$ be a certain optimization objective of the group i and assume that smaller $\mathcal{F}_i$ stands for better performance. We constitute fairness constraints as:

$$\sum_{i=1}^{T}\mathbf{max}\left(\mathcal{F}_i - \bar{\mathcal{F}}, 0\right) < c, \tag{13}$$

where $\bar{\mathcal{F}}$ is the average performance of all groups, and c is a constant standing for the upper bound for the inequation. The above inequality equation states that the sum of the differences between below-average groups' performance and the average performance should be within a certain bound c.

In our probabilistic recommendation framework, we simply choose $\mathcal{F}_i$ to be the loss of group i since it indicates the reconstruction ability of the recommendation model w.r.t each group. Hence, our fairness constraint is specified as:

$$C^{fair}(\boldsymbol{\theta}) = \left\{\sum_{i=1}^{T}\mathbf{max}\left(\mathcal{J}_i(\boldsymbol{\theta}) - \bar{\mathcal{J}}(\boldsymbol{\theta}), 0\right) - c < 0\right\} \tag{14}$$

So far, we have described our recommendation model and finished the formulation of multi-objectives and fairness constraints. Next, we proceed to develop an optimization algorithm for our fairness-constrained multi-objective problem to find out a constrained Pareto optimal solution.

4.5 Fairness-constrained multi-objective optimization

Previously in Sec. 3, we have introduced a gradient-based update rules (see Eq. 3, 4) to solve a Pareto optimal solution for unconstrained MOO problems. Interestingly, this update rule is equivalent to minimize a linear scalarization loss: $\sum_{i=1}^{T} w_i^{\tau}\mathcal{J}_i(\boldsymbol{\theta}^{\tau})$. An important distinction from the linear scalarization formulation is that the weight assignments w_i^{τ} are determined dynamically and adaptively during training. From this point of view, recall that in our problem, we have additional fairness constraints. Then after obtain w_i^{τ} by solving Eq. 3 at each step τ, we can rewrite the constrained MOO problem as the following constrained single-objective problem:

Algorithm 1 Updating rules for constrained MOO.

Initialize the model parameters $\boldsymbol{\theta}$ and initialize $\boldsymbol{\lambda} = \mathbf{0}$.
Set the learning rate $\eta_{\boldsymbol{\theta}}, \eta_{\boldsymbol{\lambda}}$, and a threshold ϵ used for stopping the optimization.
repeat
  Decide the activated set $\tilde{\mathcal{I}}(\boldsymbol{\theta}^{\tau})$
  Calculate the gradient vectors $\{\nabla\mathcal{J}_i(\boldsymbol{\theta}^{\tau})\}_{i=1}^{T}$
  Obtain w_i by solving Eq. 3
  Calculate $\boldsymbol{d}_{\boldsymbol{\theta}}^{\tau}$ by Eq. 17 and update $\boldsymbol{\theta}^{\tau+1} = \boldsymbol{\theta}^{\tau} + \eta_{\boldsymbol{\theta}}\boldsymbol{d}^{\tau}$
  Calculate the gradient/sub-gradient vectors $\nabla_{\boldsymbol{\lambda}}$ w.r.t $\mathcal{L}(\boldsymbol{\theta}^{\tau}, \boldsymbol{\lambda}^{\tau})$ and update $\boldsymbol{\lambda}^{\tau+1} = \left[\boldsymbol{\lambda}^{\tau+1} + \eta_{\boldsymbol{\lambda}}\nabla_{\boldsymbol{\lambda}})\right]_{+}$. Herein, the function $[x]_{+}$ projects x into $\mathbb{R}^{+}$.
until $\|\boldsymbol{d}_{\boldsymbol{\theta}}^{\tau}\|_2 < \epsilon$ or maximum iteration limit
return $\boldsymbol{\theta}^{\star}$

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \sum_{i=1}^{T} w_i^{\tau}\mathcal{J}_i(\boldsymbol{\theta}) \\ &s.t.\ \ C_j^{fair}(\boldsymbol{\theta}) \le 0, j = 1, \cdots, C. \end{aligned} \tag{15}$$

Herein, we assume there are C fairness constraints so that it works in more general situations. The above problem is equivalent to the following problem:

$$\min_{\boldsymbol{\theta}}\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}) := \sum_{i=1}^{T} w_i^{\tau}\mathcal{J}_i(\boldsymbol{\theta}) + \sum_{j=1}^{C}\lambda_j C_j^{fair}(\boldsymbol{\theta}) \tag{16}$$

where $\boldsymbol{\lambda}$ is a C-dimensional non-negative vector of Lagrange multipliers. Therefore, we develop a constrained MOO algorithm consisting of the following two phases. The first phase is to apply stochastic gradient descent to update model parameters $\boldsymbol{\theta}$. We first solve dynamic weight assignments $w_1^{\tau}, \cdots, w_T^{\tau}$ by problem Eq. 3. To solve it efficiently, we adopt the methods proposed in [35]. Then the model parameters are updated along the direction:

$$\boldsymbol{d}_{\boldsymbol{\theta}}^{\tau} = -\sum_{i=1}^{T} w_i^{\tau}\nabla\mathcal{J}_i(\boldsymbol{\theta}^{\tau}) - \sum_{j\in\tilde{\mathcal{I}}(\boldsymbol{\theta}^{\tau})}\lambda_j\nabla C_j^{fair}(\boldsymbol{\theta}^{\tau}). \tag{17}$$

Herein, $\tilde{\mathcal{I}}(\boldsymbol{\theta}^{\tau})$ is the index set of fairness constraints that is activated at solution $\boldsymbol{\theta}$, i.e.,

$$\tilde{\mathcal{I}}(\boldsymbol{\theta}^{\tau}) = \left\{j \,|\, C_j^{fair}(\boldsymbol{\theta}^{\tau}) > 0, 1 \le j \le C\right\}.$$

The second phase is to deal with the max subproblem over $\boldsymbol{\lambda}$. We adopt the projected gradient ascend method to update it. The overall proposed constrained MOO algorithm is concluded in the Algorithm 1.

5 EXPERIMENTS

In this section, we first synthesize user-item interactions with simulated biases and conduct experiments on the synthetic interactions to see how different user groups perform in the existence of the

Table 1: Statistics of both simulated datasets and real-world datasets. P and O are short for population bias and observation bias, respectively.

Dataset	#users	#items	#groups	sparsity (%)
Synthetic (P)	[10, 100]	197	2	[20.87, 12.30]
Synthetic (O)	[10, 10]	227	2	[13.63, 18.89]
Netflix	[3653, 3829, 3713, 3805]	3979	4	[.18, .40, .83, 2.33]

two biases and can our method improve the performance of disadvantaged groups? We then conduct experiments on two real-world datasets to see the effectiveness of our method in more complicated real-world recommendation scenarios.

5.1 Implementation details

Our proposed algorithms are implemented with TensorFlow[1]. We choose NeuMF [14] and DisRepre [25] as our backbone models, respectively. For NeuMF, we adopt the authors' hyperparameters. For the DisRepre backbone (details are attached in the supplementary materials), on a synthetic dataset, we set the hyper-parameter $d = 10$ since the synthesized records are generated from latent embeddings with $d = 10$. For real-world datasets, we validate d from a range of $\{50, 60, 70, 80, 100, 120, 150\}$ and finally adopt $d = 100$. We validate the number of cluster K from $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and finally select $K = 1$. (The reason that $K > 1$ does not make a significant improvement is perhaps because that Netflix contains only videos. A user might exhibit diversified interests among multiple categories of items, such as among videos, music and clothes, etc, but show intent interests within videos.) With the help of the validation set, the standard deviation of the prior σ_0 is set to 0.075 and the temperature factor ρ is set to 0.1. We employ ReLU [13] as the activation function and dropout technique [39] in the shallow neural network with a dropout rate of 0.5. We adopt Adam [18] optimizer with a learning rate of 0.001 since it performs well on all the experiments.

The upper bound value c is determined by first calculating $\max(\mathcal{J}_i - \bar{\mathcal{J}}), i = 1, \cdots, T$ (denoted as $\max^\star$) on the validation dataset after several epochs of training with $c = \inf$, and then search from $\{0.2, 0.4, 0.6, 0.8, 1.0\} \cdot \max^\star$. Since we have multiple evaluation metrics, we choose Recall@30 as the criterion to tune hyper parameters for simplicity.

5.2 Evaluation Metrics

We choose the averaged ranking-based metrics Recall@k (short as R@k) and HitRate@k (short as HR@k)of all users as accuracy-related metrics. Accordingly, we calculate the standard variations of the two metrics over different user groups to evaluate fairness (roughly represented by R@k-std and HR@k-std, respectively). We choose $k = 20, 30$, respectively. Besides, we plot the profile of ranked-based metrics over different groups to see what are the performance differences among different groups and how our method performs on those disadvantaged groups.

[1] https://www.tensorflow.org

5.3 Competitors

We conduct the following experiments to compare with our constrained multi-objective optimization algorithms on both synthetic data and real-world datasets.

- Overall (U). This stands for the conventional practices that optimize an overall loss of all users without any constraints (U is for Unconstrained).
- Overall (C). This method optimizes an overall loss of all users with fairness constraints. Many fairness-pursued models [29] adopt this scheme to seek fairness. We adopt the algorithm proposed by [1, 7] to solve the constrained optimization. The high-level idea of their algorithm is to play a min-max game, where one player minimizes over the model parameters, and the other player maximizes over the Lagrange multipliers of the constraint functions. We relax the constraints with a hinge-based surrogate loss for the constraints.
- Overall (R). This scheme adds a fairness regularization term to the overall loss and is adopted by many fairness-pursued models[3, 40, 41]. We search the weight of the regularization term and report the best ones.
- MOO (U). This competitor is the unconstrained multi-objective optimization method. We adopt the efficient algorithm proposed by [35] i.e., Eq. 3 and Eq. 4 to solve it.

We compare the above algorithms with our proposed algorithm on two recommendation models, i.e., NeuMF [14] and DisRepre [25], roughly representing deep learning-based CF models and the VAE-based CF models, respectively.

5.4 Synthetic data

We generate user-item interactions with a simulated recommendation scenario similar to the proposed generative recommendation model. Let the dimension of latent embeddings $d = 10$, we assume that there are two clusters of item embeddings, one is drawn from a normal distribution, $\mathcal{N}(-0.3\boldsymbol{e}, \boldsymbol{I})$ and the other is from a normal distribution $\mathcal{N}(0.3\boldsymbol{e}, \boldsymbol{I})$, where $\boldsymbol{e} = [1, 1, \cdots, 1]^\top \in \mathbb{R}^{10}$ is a vector with 1 in all entries and $\boldsymbol{I} \in \mathbb{R}^{10\times 10}$ is an identity matrix. We assume there are two user groups. The latent embeddings of users in Group 1 are generated from $\mathcal{N}(-0.3\boldsymbol{e}, \boldsymbol{I})$, while that of users in Group 2 are generated from $\mathcal{N}(0.3\boldsymbol{e}, \boldsymbol{I})$.

To introduce biases in the synthetic data, we simulate two typical biases, namely, population bias [26] and observation bias [41]. To simulate population bias, we assume Group 1 has 10 users and Group 2 has 100 users. We compute the inner-products between all user embedding and item embedding as the scores of users to items. For each user, we select the top 30 items ranked by the scores as positive ones. The reason for creating two item clusters is to build competitive sources for two user groups. For those items that are interacted by both user groups, optimizing the performance of users in Group 1 tends to pull those items close to the embedding distribution of users in Group 1, while optimizing the performance of users in Group 2 does the opposite things. Note that the clusters here do not lead to cluster-specific preference patterns, so k is also set to 1 on synthetic data.

To simulate observation bias, we assume users in Group 1 and Group 2 have different tendencies to interact with items belonging to cluster 1 and cluster 2. Specifically, for users in Group 1, we

Table 2: Performance comparisons on synthetic data with simulated population biases.

Backbone	Methods	R@20↑	HR@20↑	R@30↑	HR@30↑	R@20-std↓	HR@20-std↓	R@30-std↓	HR@30-std↓
NeuMF	overall (U)	0.506	0.448	0.590	0.545	0.025	0.022	0.021	0.024
	overall (C)	0.497	0.450	0.594	0.545	0.021	0.022	0.021	0.028
	overall (R)	0.510	0.455	0.593	0.551	0.024	0.030	0.018	0.024
	multi-group (U)	0.506	0.455	0.595	0.549	0.030	0.026	0.025	0.025
	ours	0.499	0.451	0.596	0.548	0.024	0.024	0.018	0.018
DisRepre	overall (U)	0.538	0.479	0.634	0.579	0.019	0.023	0.022	0.019
	overall (C)	0.539	0.486	0.637	0.580	0.017	0.024	0.020	0.026
	overall (R)	0.541	0.488	0.640	0.583	0.018	0.021	0.019	0.019
	multi-group (U)	0.541	0.485	0.636	0.579	0.020	0.021	0.025	0.022
	ours	0.545	0.486	0.640	0.583	0.008	0.012	0.021	0.017

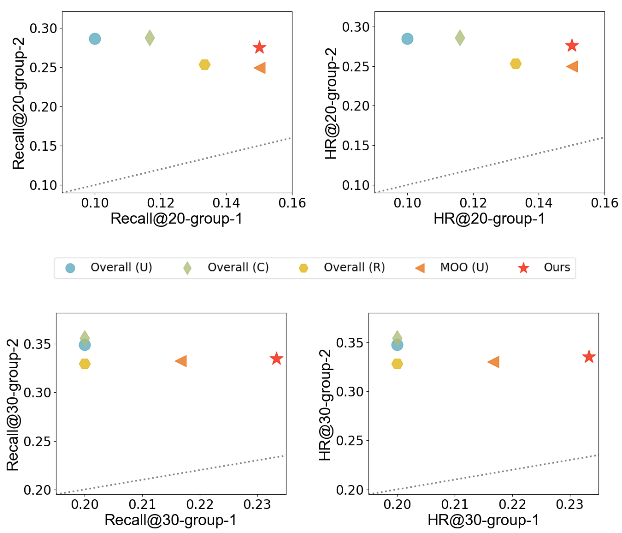


Figure 1: Group performance profile on synthetic data with simulated population biases. Different colors and shapes stand for different methods. The dotted black line stands for competitive performance of two groups. The more upper-right, the higher recommendation performance. The closer to the dotted black line, the more balanced is the performance of the two groups.

assume that 90% of the interacted items come from item cluster 1 and the other 10% comes from item cluster 2. For users in Group 2, we assume half of the interacted items come from item cluster 1 and the other half comes from item cluster 2. We filter out the items that are not interacted by any user and record the statistics of the synthetic datasets in Tab. 1. The generated interactions are then randomly split into training/validation/test splits by a ratio of 0.6/0.2/0.2.

Due to the page limit, we only include the results on synthetic data with simulated population bias and attach those with observation biases in the supplementary materials.

5.4.1 Results analysis. The overall accuracy and fairness metrics on synthetic datasets are listed in Tab. 2. To further see how models perform on different user groups, we go deeper into the individual performance of each user group, as illustrated in the performance profile in Figure 1. Herein, we simply report the profile results on the DisRepre backbone model.

From the figure, we see that Group 1 stays disadvantaged compared to Group 2 in the existence of simulated biases. In the existence of such simulated bias, users belonging to Group 1 have fewer collaborators that share similar preferences. Therefore, users in Group 1 tend to be under-represented.

From Tab. 2, we see that our method achieves comparable or better results on all of the metrics. While the improvements are not significant, we stress that it is more crucial to look deeper into the individual performance on each user group, especially for those disadvantaged user groups, saying, the performance on Group 1 here. By examining the performance profile over each user group, we see that all methods obtain similar performance on Group 2. However, the performance significantly differs on Group 1, disadvantaged group. Clearly, our method appears in the most upper-right on the figures, indicating that our method achieves the best performance on the disadvantaged group in most cases while not hurting the advantaged group much.

This improvement might attribute to that we treat each user group as an objective and avoid the gradient information being dominated by those users with a large sample frequency. Due to the MOO scheme, the weights w_i^τ of each objective $\mathcal{J}_i$ are determined dynamically and adaptively in each iteration. This is equivalent to assigning the adaptive weights for different user groups, instead of evenly distributed weights. Adaptively assigning different weights to different user groups may help in the sense that it adaptively changes the importance of different users during training to improve over-fitting on the advantaged groups and under-fitting on the disadvantaged groups.

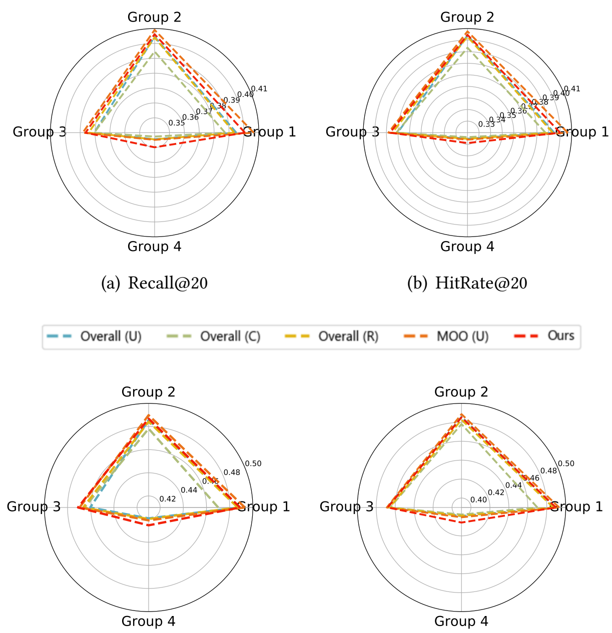
5.5 Real-world dataset

We also conduct empirical evaluations on a well-known real-world dataset for RS, i.e., Netflix Prize[2]. Netflix dataset [2] comes from the famous open competition Netflix Prize and contains ratings ranging from 1 to 5. Since this dataset is very large, we randomly sample a subset of it, whose statistics after preprocessing are shown in Tab. 1. We keep the interactions whose rating score is higher

[2]https://www.kaggle.com/netflix-inc/netflix-prize-data

Table 3: Performance comparisons on Netflix dataset.

Backbone	methods	R@20↑	HR@20↑	R@30↑	HR@30↑	R@20-std↓	HR@20-std↓	R@30-std↓	HR@30-std↓
NeuMF	overall (U)	0.329	0.305	0.411	0.383	0.018	0.023	0.019	0.027
	overall (C)	0.306	0.296	0.385	0.372	0.015	0.018	0.018	0.029
	overall (R)	0.330	0.306	0.413	0.384	0.017	0.024	0.020	0.029
	multi-group (U)	0.337	0.310	0.422	0.390	0.019	0.027	0.023	0.032
	ours	0.305	0.295	0.387	0.371	0.012	0.016	0.013	0.018
DisRepre	overall (U)	0.382	0.350	0.464	0.428	0.023	0.031	0.028	0.036
	overall (C)	0.376	0.347	0.458	0.424	0.024	0.031	0.025	0.033
	overall (R)	0.381	0.350	0.463	0.427	0.023	0.030	0.027	0.037
	multi-group (U)	0.386	0.352	0.467	0.429	0.026	0.033	0.029	0.037
	ours	0.383	0.350	0.464	0.427	0.024	0.032	0.027	0.036



(a) Recall@20 (b) HitRate@20 (d) Recall@30 (e) HitRate@30

Figure 2: Group performance profile on Netflix.

than 3 as positive interactions and only keep those users with more than 5 interaction records.

Due to the lack of user attributes, we divide users by users' interaction numbers which reflect whether a user is active or not. We take 1/4, 2/4 and 3/4 quantiles of the number of interaction records, respectively, as the criterion to divide users into 4 groups. The specific group statistics are listed in Tab. 1. We then randomly split each user's interactions into training/validation/testing splits by a ratio of 80%, 10%, 10%, respectively.

5.5.1 Results analysis. We record the experimental results on Netflix in Tab. 3. To further see how models perform on different user groups, the individual ranking-based accuracy of each group on the DisRepre backbone model is plotted in Figure 2, respectively. From the table, we see that our method obtains smaller accuracy variations across user groups or maintaining top recommendation performance. It is reasonable to observe that a method could not perform well on both accuracy-related metrics and fairness-related metrics since the two objectives are not totally consistent. Again, it is more significant to look deeper into the performances of different user groups, especially for those disadvantaged groups. From the figures, we see that our method produces solutions that dominate the other solution w.r.t each user group in most of the cases. Especially, our method could improve the performance of the disadvantaged groups, i.e. Group 4 on Netflix, while assuring stable recommendation performance for the other user groups.

6 CONCLUSION

In this paper, we study the trade-off between overall recommendation performance and fairness across user groups. Instead of optimizing an overall objective with additional fairness constraints or regularizations defined by equality of performance across groups, we formulate a fairness-constrained multi-objective optimization problem to treat each subgroup equivalently as an objective. To solve this problem, a gradient-based constrained MOO algorithm is proposed to seek a proper Pareto optimal solution for the performance trade-off. The algorithm is equivalent to assign adaptive weights to each user group automatically during training. This ensures that the imbalanced subgroup sample frequency does not affect the gradient information. Extensive experiments on synthetic and real-world datasets show that our approach could help improve the performance of the disadvantaged groups while not damaging the overall performance.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102104, in part by National Natural Science Foundation of China: 61931008, 61620106009, 61836002 and 61976202, in part by the Fundamental Research Funds for the Central Universities, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDB28000000, in part by the National Postdoctoral Program for Innovative Talents under Grant BX2021298 and in part by MindSpore, which is a new deep learning computing framework [3].

[3]https://www.mindspore.cn/

REFERENCES

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. 2018. A Reductions Approach to Fair Classification. In *International Conference on Machine Learning*, Vol. 80. 60–69.

[2] Robert M. Bell and Yehuda Koren. 2007. Lessons from the Netflix prize challenge. *SIGKDD Explorations* 9, 2 (2007), 75–79.

[3] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2212–2220.

[4] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *ACM SIGIR Conference on Research & Development in Information Retrieval*. 405–414.

[5] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with Fairness Constraints. In *International Colloquium on Automata, Languages, and Programming*. 28:1–28:15.

[6] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In *International Conference on Machine Learning*. 793–802.

[7] Andrew Cotter, Heinrich Jiang, Maya R. Gupta, Serena Wang, Taman Narayan, Seungil You, and Karthik Sridharan. 2019. Optimization with Non-Differentiable Constraints with Applications to Fairness, Recall, Churn, and Other Goals. *Journal of Machine Learning Research* 20, 172 (2019), 1–59.

[8] Kalyanmoy Deb. 2001. *Multi-objective Optimization using Evolutionary Algorithms.* Wiley.

[9] Jeanantoine Desideri. 2012. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique* 350, 5-6 (2012), 313–318.

[10] Jörg Fliege and Benar Fux Svaiter. 2000. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research* 51, 3 (2000), 479–494.

[11] Jörg Fliege and A. Ismael F. Vaz. 2016. A Method for Constrained Multiobjective Optimization Based on SQP Techniques. *SIAM Journal on Optimization* 26, 4 (2016), 2091–2119.

[12] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2221–2231.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *International Conference on Computer Vision*. 1026–1034.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *International World Wide Web Conferences*. 173–182.

[15] Yifan Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *IEEE International Conference on Data Mining*. 263–272.

[16] Nathan Kallus and Angela Zhou. 2019. The Fairness of Risk Scores Beyond Classification: Bipartite Ranking and the XAUC Metric. In *Advances in Neural Information Processing Systems*. 3433–3443.

[17] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7482–7491.

[18] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

[19] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

[20] Y. Koren, R. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[21] Y.-J. Leng, Q. Lu, and C.-Y. Liang. 2014. Survey of recommendation based on collaborative filtering. *IEEE Transactions on Pattern Recognition and Artificial Intelligence* 27, 8 (2014), 720–734.

[22] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *International World Wide Web Conferences*. 689–698.

[23] Xiao Lin, Hongjie Chen, Changhua Pei, Fei Sun, Xuanji Xiao, Hanxiao Sun, Yongfeng Zhang, Wenwu Ou, and Peng Jiang. 2019. A Pareto-Efficient Algorithm for Multiple Objective Optimization in e-Commerce Recommendation. In *ACM Conference on Recommender Systems*. 20–28.

[24] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. 2019. Pareto Multi-Task Learning. In *Advances in Neural Information Processing Systems*. 12037–12047.

[25] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *Advances in Neural Information Processing Systems*. 5712–5723.

[26] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A Survey on Bias and Fairness in Machine Learning. *arXiv preprint arXiv:1909.03388* (2019).

[27] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a Fair Marketplace: Counterfactual Evaluation of the trade-off between Relevance, Fairness & Satisfaction in Recommendation Systems. In *ACM International Conference on Information and Knowledge Management*. 2243–2251.

[28] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. In *the International ACM SIGIR conference on research and development in Information Retrieval*. 429–438.

[29] Harikrishna Narasimhan, Andrew Cotter, Maya R. Gupta, and Serena Wang. 2020. Pairwise Fairness for Ranking and Regression. In *AAAI Conference on Artificial Intelligence*. 5248–5255.

[30] Sebastian Peitz and Michael Dellnitz. 2018. Gradient-based multiobjective optimization with uncertainties. In *NEO 2016*. 159–182.

[31] Fabrice Poirion, Quentin Mercier, and Jean-Antoine Désidéri. 2017. Descent algorithm for nonsmooth stochastic multiobjective optimization. *Computational Optimization and Applications* 68, 2 (2017), 317–331.

[32] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning*. 1278–1286.

[33] Michael Roy. 2017. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. *College & Research Libraries* 78, 3 (2017), 403–404.

[34] S. Schäffler, R. Schultz, and K. Weinzierl. 2002. Stochastic Method for the Solution of Unconstrained Vector Optimization Problems. *Journal of Optimization Theory and Applications* 114, 1 (2002), 209–222.

[35] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. In *Advances in Neural Information Processing Systems*. 525–536.

[36] Dimitris Serbos, Shuyao Qi, Nikos Mamoulis, Evaggelia Pitoura, and Panayiotis Tsaparas. 2017. Fairness in Package-to-Group Recommendations. In *International World Wide Web Conferences*. 371–379.

[37] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2219–2228.

[38] Ashudeep Singh and Thorsten Joachims. 2019. Policy Learning for Fairness in Ranking. In *Annual Conference on Neural Information Processing Systems 2019*. 5427–5437.

[39] Nitish Srivastava, Geoffrey E. Hinton, A. Krizhevsky, Ilya Sutskever, and R. Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[40] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In *ACM Conference on Recommender Systems*. 107–115.

[41] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *Advances in Neural Information Processing Systems*. 2921–2930.

[42] Sirui Yao and Bert Huang. 2017. New fairness metrics for recommendation that embrace differences. *arXiv preprint arXiv:1706.09838* (2017).

[43] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *International Conference on Information and Knowledge Management*. 1569–1578.

[44] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. 2019. Advances in Variational Inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2019), 2008–2026.

[45] Qingfu Zhang and Hui Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 712–731.

[46] Eckart Zitzler. 1999. *Evolutionary algorithms for multiobjective optimization: Methods and applications.* Vol. 63.