Balancing Fairness in Multi-stakeholder Recommendation via Multi-Objective Optimization

Haolun Wu

School of Computer Science
McGill University
Montréal, Québec, Canada

April 15, 2021

A thesis presented for the degree of Master of Computer Science

Abstract

Fairness in machine learning has been investigated a lot in academia and industry; however, the fairness problem in recommendation has just been noticed recently and is still under-researched, especially the fairness of multi-stakeholder. Generally, two of the most important stakeholders in a modern recommendation system are (i) producers, represented by goods and services (e.g., movies on Netflix and products on Amazon) and (ii) customers who pay for them. Traditionally, recommendation services have focused on maximizing consumer satisfaction by tailoring the results according to the personalized preferences of individual consumers. However, such a consumer-centric design is harmful to the exposure of producers in a recommendation system, thus being regarded as "unfair".

Here, we first offer comprehensive background on the recommendation systems, followed by the introduction of fairness in recommendation from the view of different stakeholders, since they have different requirements on fairness even in the same system. Further, we consider balancing the multi-stakeholder fairness in a two-sided

marketplace, focusing on consumers and producers. We propose a fairness-aware recommendation framework by using multi-objective optimization (MOO), Multi-FR, to adaptively balance the objectives between consumers and producers. In particular, Multi-FR adopts the multi-gradient descent to generate a Pareto set of solutions, where the most appropriate one is selected from the Pareto set. In addition, four fairness metrics/constraints are applied to make the recommendation results on both the consumer and producer side fair. We extensively evaluate our model on three real-world datasets, comparing with grid-search methods and using a variety of performance metrics. The experimental results demonstrate that Multi-FR can improve the recommendation fairness on both the consumer and producer side with little drop in recommendation quality, also outperforming several state-of-the-art fair ranking approaches. This approach is applicable to balance fairness with respect to any number of stakeholders.

Abrégé

L'équité dans l'apprentissage automatique a fait l'objet de nombreuses recherches dans les universités et l'industrie; cependant, le problème d'équité dans la recommandation vient d'être remarqué récemment et est encore sous-étudié, en particulier l'équité de multi-parties prenantes. En règle générale, deux des parties prenantes les plus importantes dans un système de recommandation moderne sont (i) les producteurs, représentés par des biens et services (par exemple, des films sur Netflix et des produits sur Amazon) et (ii) les clients qui paient pour eux. Traditionnellement, les services de recommandation se sont concentrés sur la maximisation de la satisfaction des consommateurs en adaptant les résultats en fonction des préférences personnalisées des consommateurs individuels. Cependant, une telle conception centrée sur le consommateur est préjudiciable à l'exposition des producteurs dans un système de recommandation, et est donc considérée comme "injuste".

Ici, nous proposons d'abord un contexte complet sur les systèmes de recommandation, suivi de l'introduction de l'équité dans la recommandation du point

de vue de différentes parties prenantes, car elles ont des exigences différentes en matière d'équité, même dans le même système. De plus, nous envisageons d'équilibrer l'équité multipartite dans un marché à deux faces, en se concentrant sur les consommateurs et les producteurs. Nous proposons un cadre de recommandation soucieux de l'équité en utilisant l'optimisation multi-objectifs (MOO), Multi-FR, pour équilibrer de manière adaptative les objectifs entre les consommateurs et les producteurs. En particulier, Multi-FR adopte la descente multi-gradient pour générer un ensemble de solutions Pareto, où la plus appropriée est sélectionnée dans l'ensemble de Pareto. De plus, quatre mesures/contraintes d'équité sont appliquées pour rendre équitables les résultats de la recommandation du côté du consommateur et du producteur. Nous évaluons en profondeur notre modèle sur trois ensembles de données du monde réel, en comparant avec des méthodes de recherche par grille et en utilisant une variété de mesures de performance. Les résultats expérimentaux démontrent que Multi-FR peut améliorer l'équité des recommandations à la fois du côté des consommateurs et des producteurs avec une faible baisse de la qualité des recommandations, surpassant également plusieurs approches de classement des foires de pointe. Cette approche est applicable pour équilibrer l'équité à l'égard d'un nombre quelconque de parties prenantes.

Acknowledgements

Firstly, I would like to express my sincere appreciation to my supervisor Professor Xue Liu for his continuous support of my MSc study and this thesis work. His consistent guidance on the research and this work has made me through many challenges. Prof. Liu's patience, encouragement, and broad knowledge inspire me to continue my research career and explore more in this field.

I would also like to give my special thank to Chen Ma, a senior Ph.D. student in my lab, for providing many suggestions on technical details. His detailed advice offered me many valuable insights into both the theoretical model and the simulation experiments. I also appreciate other members in our CPS Lab, Fuyuan Lyv, Sirui Song, Can Chen, who always share their precious thoughts and discuss with me on broad research topics during my master study. Without these excellent peers, I cannot improve myself so much.

I would also express my gratitude to my parents for their continuous love and support throughout my master's study. Most importantly, thanks to my wife who supports me during the past two years. Best wishes to us for our upcoming Ph.D. studies at McGill.

Contents

List of Figures

List of Tables

List of Acronyms

BPR	Bayesian Personalized Ranking.
CF	Collaborative Filtering.
DCG	Discounted Cumulative Gain.
GPU	Graphics Processing Unit.
KKT	Karush-Kuhn-Tucker.
MF	matrix factorization.
MGDA	Multiple Gradient Descent Algorithm.
ML	Machine Learning.
MOO	Multi-Objective Optimization.
MSE	mean squared error.
NDCG	Normalized Discounted Cumulative Gain.
NGCF	Neural Graph Collaborative Filtering.
ReLU	rectified linear unit.
RS	Recommendation System.

SOTA State-Of-The-Art.

SVD singular value decomposition.

WRMF Weighted Regularized Matrix Factorization.

Chapter 1

Introduction

This chapter mainly summarizes the background of recommendation systems and the fairness problem in the recommendation. We first detail the motivation and widely used approaches in recommendation systems. Then we introduce the topic of the fairness problem in the recommendation scenario with respect to multiple stakeholders. In the end, we list the contributions in this thesis and close this chapter by summarizing the organization of this thesis.

1.1 Background on Recommendation Systems

With the rapid development of this information age, the explosive growth in the amount of digital information and the number of Internet users have created a big challenge of information overload which hinders real-time access to items of interest on the Internet.

This has increased the demand for recommendation systems more than ever before. Recommendation systems are information filtering systems that deal with the problem of information overload [Konstan and Riedl, 2012] by filtering significant information fragment out of a large amount of dynamically generated information according to user's preferences, interest, or observed behaviour about item [Chenguang Pan and Wenxin Li, 2010].

The goal of recommendation systems is to provide users with personalized recommendations for products that they would like. Typically, the development of recommendation systems makes it easier for users to speed up the searches and access items they never searched for but may still prefer in real time. Many e-commerce companies have been using recommendation systems to find out consumers' real preference followed by recommending products to them. As such, companies can increase sales by offering personalized recommendations and enhancing customer experience. Sites like Amazon and Youtube generate different suggested playlists and make video recommendations for each user, e.g., as shown in Fig. 1.1. Moreover, companies can retain customers by sending out emails with the link to suggestions of other items that they might like. These systems make effective use of the knowledge available and may be viewed as a rank searching system, where the input query may be a set of users and auxiliary information, while the output may be a ranked list of items. Given a query, a system needs to find the relevant items and rank the items based on

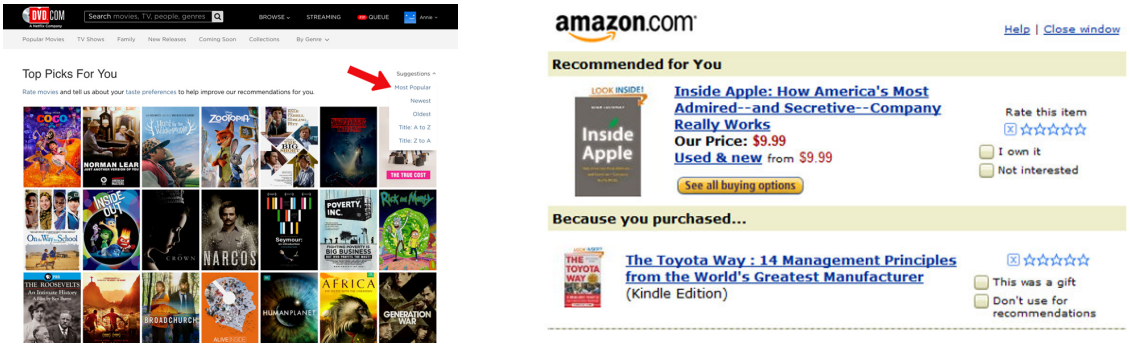


Figure 1.1: Examples of Recommendation Systems

certain objectives, such as purchases or clicks [Konstan and Riedl, 2012].

Data: Explicit feedback and Implicit feedback. In order to build a recommendation system, data is one of the most important prerequisites. Input data of the recommendation system are often placed in a matrix, where one dimension represents the users and the other represents the items. There are basically two input data. One is the explicit feedback, which includes explicit values by users regarding their interest in products. For instance, Netflix collects star ratings for movies, here, ratings are explicit feedback. However, explicit feedback is not always available in real scenarios, since usually users only rate a small number of items. Thus, most recommendation systems infer user preferences through the other type of data, the implicit feedback. In implicit feedback, the user's preference is reflected by observing behaviour such as purchase history, browsing history or search patterns [Amatriain et al., 2009a], which is encoded with 0 or 1. A one represents a user observes/likes an item, where a zero represent a

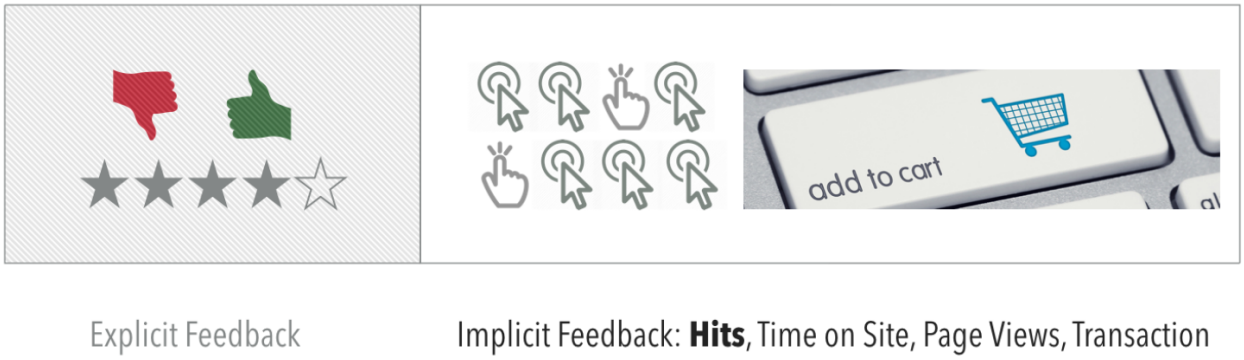


Figure 1.2: Examples for explicit feedback and implicit feedback. Source from `https://jinyi.me/2018/06/Recommendation-System-Miscellus/`.

user dislikes or does not observe an item. Examples for these two types of feedbacks are shown in Fig. 1.2.

There are similarities and differences between these two types of feedback. Both suffer from noise [Amatriain et al., 2009b, Anand et al., 2007, Hu et al., 2008] and are sensitive to the user's context, albeit not to the same extent. In terms of differences, as aforementioned, explicit feedback is not easy to obtain whereas implicit feedback is abundant. The most notable difference is that explicit feedback can be positive or negative, while implicit feedback is only sure about the positive, since the negative implicit feedback may have two interpretations: (1) the user does not observe it or (2) the user does not like it. Thus, the explicit feedback is generally more accurate than implicit feedback in representing the user's real preference. Furthermore, explicit feedback tends to concentrate on either side of the rating scale, as users are more likely to express their preferences if they feel strongly for or against an item [Amatriain et al., 2009a]. We

	Implicit Feedback	Explicit Feedback
Context-Sensitive	Yes	Yes
Accuracy	Low	High
Abundance	High	Low
Expressivity	Positive	Positive and Negative
Measurement	Relative	Absolute

Table 1.1: The characteristics of explicit and implicit feedback.

summarize the characteristics of explicit and implicit feedback in Table 1.1.

Common approaches: content-based and collaborative filtering. Existing methods for recommendation systems can roughly be categorized into three classes [Bobadilla et al., 2013]: content-based methods, collaborative filtering-based methods, and hybrid methods, in which the first two are at the core. Content-based methods focus on the similarity of item attributes, while collaborative filtering approaches are based on the similarity of user-item interactions. An illustration for showing the difference between these two approaches is in Fig. 1.3.

Content-based methods are domain-dependent algorithms and they emphasize more on the analysis of the attributes of items in order to generate predictions. In content-based approaches, the recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past [Bobadilla et al., 2013, Vekariya and Kulkarni, 2012]. Items that are mostly related to the positively rated items are recommended to the user. As an example, a book profile

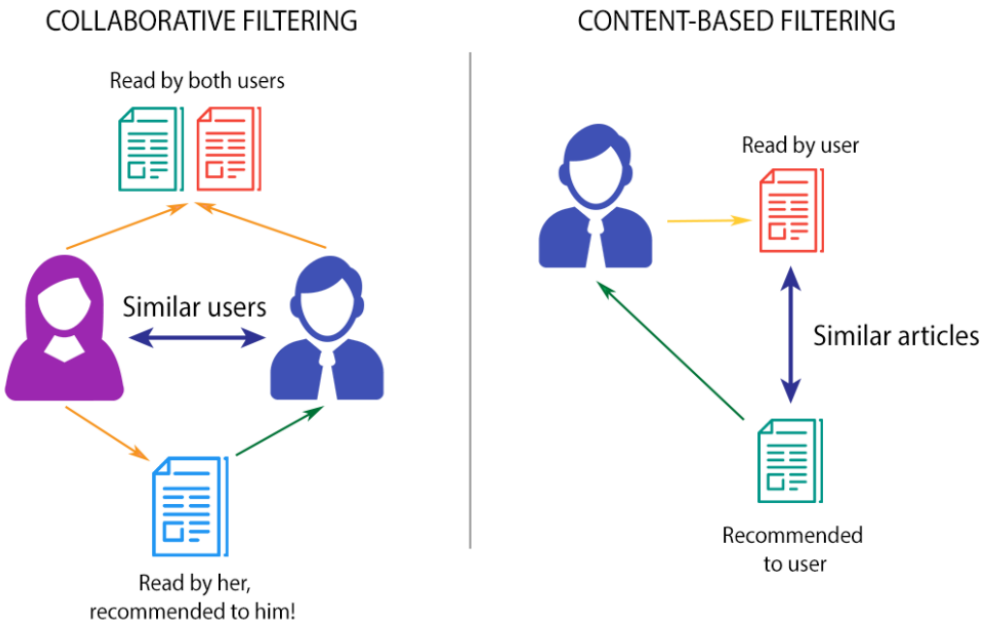


Figure 1.3: Comparison between collaborative filtering approach and content-based approach. Source from paper [Lu et al., 2013].

might include its author, its genre, its published date, etc. User profiles might include demographic information or answers provided on a suitable questionnaire. There are some other systems that also utilize user social and personal data. The hypothesis behind this method is that if a user is interested in one item in the past, he/she will be interested in other similar items in the future. Items are grouped based on the similarity of the profiles. These profiles allow programs to match users with unique items.

Content-based approaches do not need the profile of other users since they do not influence recommendations. Therefore, they have the ability to recommend new items even if there are no ratings provided by users. Also, if the user preferences change, it is still able to adjust the recommendations in a short span of time. They can manage

situations where different users do not share the same items, but only identical items according to their intrinsic features. However, the major disadvantage of this technique is the need to have a rich knowledge of the profile.

Different from the content-based methods, collaborative filtering (CF) based methods are domain-independent. It has been proved that CF is very effective for forecasting customer precedence in the choice of objects. This method is flourished in the middle of the 1990s with retail services which utilized recommendation systems and presented online, e.g. Netflix, Amazon. CF-based methods [Billsus and Pazzani, 1998] use history data or previous preferences, such as user ratings on items, without requiring users and items profiles information. They hypothesis that if one user A likes item i and another user B also likes item i, these two users may share the same interests: given user A likes item j, we may assume user B may also be interested in item j. The programs analyze interaction relationships between users and inter-dependencies among items [Koren et al., 2009]. The users will be recommended items that people with similar preferences and tastes have liked in the past.

Collaborative filtering methods can be divided into two sub-classes: memory-based CF and model-based CF.

- **Memory-based CF**: The key idea of memory-based CF approaches is that they use only information from the user-item interaction matrix and they assume no model to produce new recommendations. And the memory-based CF can still be categorized

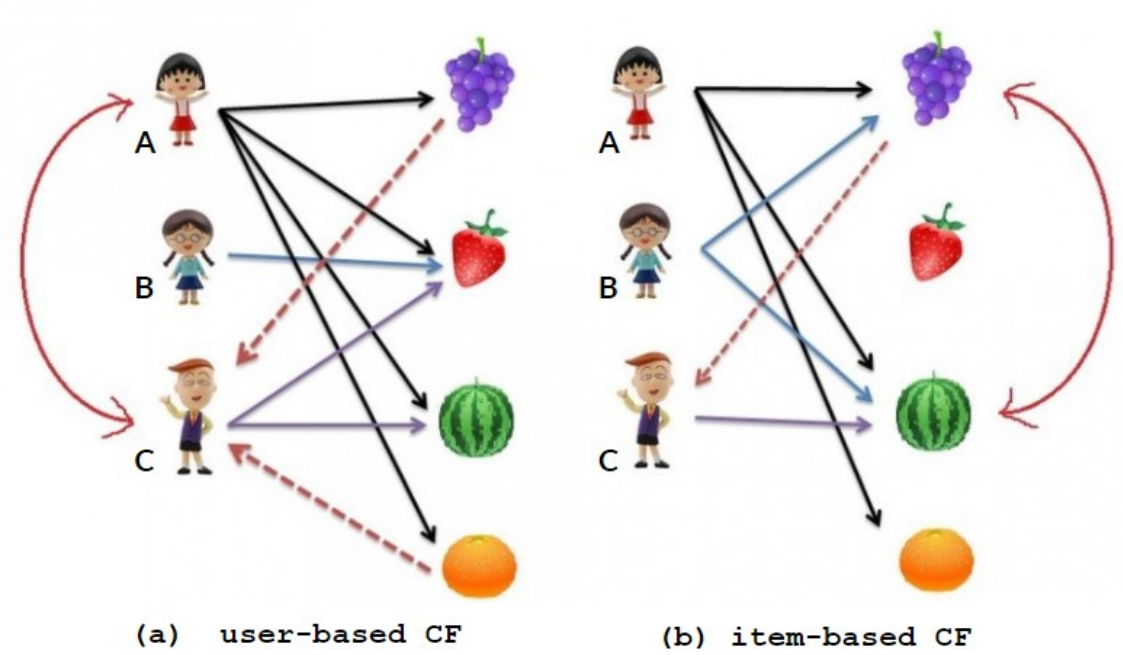


Figure 1.4: Two subclasses of memory-based CF: user-based CF and item-based CF.

into User-based CF and Item-based CF. Fig. 1.4 shows an example for these two sub-classes respectively:

- **User-based CF**: In this approach, systems cluster users who share common interests. They recommend items to a user based on the preference of users of the same neighbourhood. For example, in Fig. 1.4 (a), user A and C both like strawberry and watermelon. Systems would recommend user C grapes and oranges which user A also likes.
- **Item-based CF**: Referring to the fact that the taste of users remains constant or change very slightly, similar items are clustered based on the ratings or comments of users. Items are recommended to a user from the same

neighbourhood that a user might prefer. In Fig. 1.4 (b), grapes and watermelon are grouped to a similar neighbourhood because user A and user B both like these two. So when user C likes Watermelon, the other item from the same neighbourhood, i.e Grapes, will be recommended by item-based CF.

- **Model-based CF**: Model-based collaborative approaches only rely on user-item interactions information and assume a latent model supposed to explain these interactions. These techniques can quickly recommend a set of items for the fact that they use a pre-computed model and they have proved to produce recommendation results that are similar to neighbourhood-based recommender techniques. Examples of these techniques include Dimensionality Reduction techniques such as Singular Value Decomposition (SVD), Matrix Completion Technique, Latent Semantic methods, and Regression and Clustering.

  Specifically, for instance, matrix factorization algorithms consist in decomposing the huge and sparse user-item interaction matrix $\mathbf{M} \in \mathbb{R}^{|U| \times |I|}$ into a product of two smaller and dense matrices: a user-factor matrix $\mathbf{U} \in \mathbb{R}^{|U| \times d}$ (containing users representations) that multiplies a factor-item matrix $\mathbf{V} \in \mathbb{R}^{|I| \times d}$ (containing items representations):

$$\mathbf{M} \approx \mathbf{U} \cdot \mathbf{V}^T \tag{1.1}$$

  Then the optimization process is to learn the user and item embeddings to minimize

the reconstruction loss with a regularization term scalarized by λ:

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}(\mathbf{U}_i\mathbf{V}_j^T - \mathbf{M}_{ij})^2 + \frac{\lambda}{2}(||\mathbf{U}||_2^2 + ||\mathbf{V}||_2^2) \tag{1.2}$$

The CF approach has some major advantages over the content-based approach in that it can perform in domains where there is not much content associated with items and where content is difficult for a computer system to analyze. Also, the CF technique has the ability to provide serendipitous recommendations, which means that it can recommend items that are relevant to the user even without the content being in the user's profile [Schafer et al., 2007]. Despite the success of CF techniques, they still have several challenges. One challenge is known as the cold-start problem, which refers to a situation where a recommender does not have adequate information about a user or an item in order to make relevant predictions. The other known challenge is called data sparsity which occurs as a result of lack of enough information, that is when only a few of the total number of items available in a database are rated by users [Burke, 2002, Park et al., 2012]. This always leads to a sparse user-item matrix, inability to locate successful neighbours and finally, the generation of weak recommendations.

1.2 Fairness in Recommendation

When viewed from a sociotechnical lens, conventionally deployed machine learning systems demonstrate a range of socially problematic behaviours including algorithmic bias and misinformation. Multi-sided recommendation systems such as marketplaces, content-distribution networks, and match-making platforms require reasoning about the potential impact on several populations of stakeholders with potentially disparate and possibly conflicting objectives. As such, the potential societal implications of a recommendation algorithm must balance multiple objectives across these groups.

Two-sided and multi-stakeholder recommendation systems are increasingly powering search and discovery on platforms supporting individuals attempting to satisfy human needs such as entertainment, medical information, or income. Two of the most important stakeholders in such systems are (i) producers, represented by goods and services (e.g., movies on Netflix and products on Amazon) and (ii) consumers who pay for them. When a recommendation system systematically underperforms for certain historically disadvantaged groups, inequity can be exacerbated for both consumers (e.g., users seeking content) and producers (e.g., users producing content).

Unfortunately, state-of-the-art approaches or platforms for recommendation systems are limited in addressing these issues. Traditional user-based approaches can result in unfairness both for consumers and producers, since they mainly focused on maximizing customer satisfaction by tailoring the results according to the personalized preferences

of individual customers, largely ignoring the interest of the producers. Generally, these platforms employ various data-driven methods [Koren et al., 2009, Kurokawa et al., 2016, Desrosiers and Karypis, 2011], to estimate the relevance scores of each product-customer pair, and then recommend the top-k most relevant products to the corresponding customers. However, these methods can create a huge disparity in the exposure of the producers on real-world datasets due to the "superstar economics" [Mehrotra et al., 2018, Baranchuk et al., 2011], which is unfair for the producers and may also harm the health of the marketplace. Other approaches address either consumer *or* producer unfairness, but disregarding the sensitive attributes. In addition, although a few approaches considering both the consumer and producer fairness [Geyik et al., 2019, Patro et al., 2020], they require fine-tuning of weights for multiple objectives, which can be a limitation when fairness objectives have different numerical magnitudes, especially as the number of fairness dimensions increases or the proposed model is not able to be trained end-to-end.

1.3 Contributions

To address the aforementioned problems, we treat the multi-stakeholder recommendation as a Multi-Objective Optimization (MOO) problem. The reason we adopt MOO is that manually setting the scaling factor for each objective may not make

the solution achieve the Pareto efficient condition. Existing methods for Pareto optimization can be mainly divided into heuristic search and scalarization. Multi-objective evolutionary algorithms are popular choices in heuristic search, however, they only ensure the resulting solutions are not dominated by each other (but still can be dominated by Pareto efficient solutions) [Kim et al., 2004]. Thus they cannot guarantee Pareto efficiency. The scalarization method converts multiple objectives into a single objective with weighted summation and can achieve Pareto efficient solutions with proper scalarization [Ghane-Kanafi and Khorram, 2015]. Moreover, Pareto efficient scalarization solutions can be generated by MOO.

As such, we propose a scalable and end-to-end framework, namely *Multi-FR*, which allows to balance the weight of multiple consumer and producer fairness objectives in a two-sided marketplace without the tedious weighting parameter tuning. Along with this framework, we propose two fairness constraints on the consumer side according to gender and age attributes, as well as two fairness constraints on the producer side with respect to genres and popularities. The Bayesian Personalized Ranking (BPR) is utilized for our model training, and we apply the multi-gradient descent with the Frank-Wolf Solver [Frank and Wolfe, 1956, Sener and Koltun, 2018] for finding the Pareto optimal solution to balance multiple fairness objectives with a little drop of the recommendation quality.

Finally, we tested our approach on several real-world datasets with a variety of

fairness constraints. We also compare our *Multi-FR* framework with the grid-search strategy and investigate the trend of generated scaling factors. Our results demonstrate that *Multi-FR* can find solutions that balance fairness constraints with substantial improvements over methods using hand-crafted normalization factors. To summarize, the contributions of this work include:

- We propose a generic fairness-aware recommendation framework with multi-objective optimization, *Multi-FR*, which jointly optimizes fairness and utility for a two-sided recommendation.(**Chapter 3 & 4**)
- We treat the multi-stakeholder recommendation as a MOO problem and apply the multi-gradient descent with the Frank-Wolf Solver which enables the reach of the Pareto optimal point without hand-engineering the scaling factor on objectives. (**Chapter 3 & 4**)
- Different fairness metrics like Gini Index and Simpson's Diversity Index are utilized to measure fairness from different aspects. The results indicate that *Multi-FR* can largely improve the recommendation fairness with little drop in the accuracy. (**Chapter 5**)
- Extensive experimental results on three public benchmarks datasets and three SOTA fair ranking approaches (FOEIR, FairBandit, FairRec) demonstrate the effectiveness of our proposed framework. (**Chapter 5**)

1.4 Structure of the Thesis

The remaining chapters are organized as follows.

We first introduce the related work on two-sided fairness in the recommendation, then review the approaches on achieving fairness and the multiple objectives recommendation in Chapter 2. Then, in Chapter 3, we propose a model for achieving multi-sided fairness in recommendation through multi-objective optimization. In Chapter 4, we detail how to construct the fairness objectives on both the consumer side and the producer side in a recommendation scenario. We then show the results of our comprehensive experiments to prove the effectiveness and scalability of our proposed framework in Chapter 5. Finally, we summarize our conclusion and point out several possible aspects for future work in Chapter 6.

Our contributions in each research chapter are detailed in Section 1.3.

Chapter 2

Related Work

In this chapter, we provide a summary regarding the related studies from the following aspects: two-sided fairness in the recommendation, approaches on achieving fairness, and recommendation with multiple objectives.

2.1 Two-sided Fairness in Recommendation

Prior works in fairness, in the context of recommendation systems, consider algorithmic effects on consumers (i.e. users who seek content) and producers (i.e. users who provide content), independently or together.

For the consumer side, the fairness refers to systematic differential performance [Mehrotra et al., 2017] across users and, most often, demographic groups of users. In the context of book recommendations, Ekstrand et al. [2018a] find that standard

recommendation algorithms could result in differential performance across demographic groups. Chaney et al. [2018] demonstrate, through simulation, that feedback loops inherent in the production system could exacerbate unfairness and homogenize recommendation. Yao and Huang [2017] demonstrate that these issues can be addressed by designing fairness metrics and introducing them as learning objectives.

Instead, other works focus on the fairness on the producer side, whose fairness most often refers to systematic differential exposure [Biega et al., 2018, Singh and Joachims, 2019, Diaz et al., 2020] across content producers and, most often, groups of producers (e.g. genre, popularity). Ekstrand et al. [2018b] find that standard recommendation algorithms could result in certain demographic groups being over- or under-represented in recommendation decisions. Beutel et al. [2019] demonstrate that these issues can be addressed in production systems by defining pairwise fairness objectives and introducing them as learning objectives.

Joint optimization of consumer and producer fairness is an important property for a healthy platform. Burke et al. [2018] introduce the task of two-sided fairness and propose several methods to address it, although none directly optimize fairness metrics. Mehrotra et al. [2018] treat the two-sided fairness as a multi-objective optimization task but experiments with algorithms that either use one fairness metric as a constraint or linearly interpolates fairness objectives. The first approach does not allow a search for a Pareto-optimal solution; the second approach requires tuning of an interpolation

parameter, which can be brittle and unscalable in practice. Finally, Sühr et al. [2019] experiment with two-sided fairness in the context of ride-hailing platforms. Like other works, the two-sided objective is a linear interpolation of consumer and producer fairness metrics.

2.2 Approaches on Achieving Fairness

Motivated by the idea of constructing multiple objectives in recommendation [Jambor and Wang, 2010a, McNee et al., 2006], most works on fairness in recommendation and ranking scenario model the fairness as an extra loss as a supplement to the accuracy (utility) loss in the whole objective function [Singh and Joachims, 2019, Xiao et al., 2017, Singh and Joachims, 2018], followed by the scalarization technique. It is expected to achieve a Pareto efficient recommendation [Ribeiro et al., 2012, 2015] when multiple objectives are concerned; however, existing studies mostly depend on manually assigned weights for scalarization, whose Pareto efficiency can not be guaranteed.

Recent studies have proposed to use the adversarial learning and causal graph reasoning techniques to achieve fairness in recommendation. For instance, Beigi et al. [2020] propose an adversarial learning-based recommendation with attribute protection, which can protect users from the private-attribute inference attack while simultaneously recommending relevant items to users. Rahman et al. [2019] find biases in the

recommendations are caused by unfair graph embeddings and propose a novel fairness-aware graph embedding algorithm $Fairwalk$ to achieve the statistical parity. Bose and Hamilton [2019] combined the adversarial training with the graph representation learning together to protect sensitive features on users. They introduce an adversarial framework to enforce fairness constraints on graph embeddings. The benefits of these algorithms lie into explicitly modeling the fairness into the representation embeddings; however, the models are based on more advanced techniques and they do not consider a multi-sided fairness.

Fair Learning-to-Rank (LTR) is another popular research direction for achieving fairness in community nowadays, and several recent works have raised the question of group fairness in rankings. Zehlike et al. [2017] formulate the problem as a "Fair Top-k ranking" that guarantees the proportion of protected group items in every prefix of the top-k ranking is above a minimum threshold. Celis et al. [2018] propose a constrained maximum weight matching algorithm for ranking a set of items efficiently under a fairness constraint indicating the maximum number of items with each sensitive attribute allowed in the top positions. Most recently, some works break the parity constraints restricting the fraction of items with each attribute in the ranking, but extend the LTR methods to a large class of possible fairness constraints. For instance, Biega et al. [2018] aim to achieve amortized fairness of attention by making exposure proportional to relevance through integer linear programming. Singh and Joachims

[2018] propose a more general framework which can achieve both individual fairness and group fairness solutions via a standard linear program and the Birkhoff-von Neumann decomposition [Birkhoff, 1967].

2.3 Recommendation with Multiple Objectives

The studies on multi-objective optimization are rich and various approaches have been proposed [Deb et al., 2016]. One significant feature of the multiple objective optimization is that, usually, there does not exist a solution that satisfies all the objectives simultaneously.

Some studies have considered multiple objectives in personalized recommendation tasks [Ribeiro et al., 2012, Jambor and Wang, 2010b]. For instance, Ribeiro et al. [2012] constructe multiple objectives including accuracy, diversity, and novelty and a Pareto frontier is found to satisfy the mentioned objectives. However, the manual scalarization (grid search) is still required. Besides, there are few studies on optimizing multiple objectives in group recommendation and we are among the first to treat the two-sided fairness problem in recommendation into a multi-objective optimization perspective.

Our study expands on prior works by studying interpolation-free optimization of multi-sided fairness problems. Our method is flexible and allows multiple consumer and producer fairness definitions (e.g. for different demographic groupings) without the

need to directly reason about scaling different objectives.

Chapter 3

The Proposed Framework: Multi-FR

In a conventional recommendation system, the main aim lies in satisfying the need of users/customers. However, it has been shown in recent studies [Patro et al., 2020] that solely optimizing the satisfaction of customers may jeopardize the benefits of item providers/producers who are essential participants in two-sided markets, such as Amazon and Yelp. Thus, how to achieve personalized, satisfactory, and fair recommendation simultaneously is non-trivial.

Traditionally, these aspects are modelled as specific objectives and combined by summation with different scaling factors. However, utilizing hand-crafted scaling factors has two major drawbacks. First, these scaling factors incur tedious hyper-parameter tuning. This would cost many trials and substantial computation resources to select appropriate scaling factors, especially when the number of objectives

is huge. Second, each objective in the summed objective function may need a different magnitude of scaling values in the training process. Setting one fixed value may not dynamically balance these objectives.

To tackle the aforementioned problems, we treat the fairness-aware recommendation as a multi-objective optimization problem and propose a framework to optimize multiple objectives jointly.

3.1 Multi-Objective Optimization

A Multi-Objective Optimization Problem (MOO) is usually defined as optimizing a set of possibly conflicting objectives. Given a set of objectives, the MOO aims to find a solution that can optimize all objectives simultaneously:

$$\min_{\theta} \mathbf{L}(\theta) = \min_{\substack{\theta^c \\ \theta^{s_1},...,\theta^{s_t}}} \mathbf{L}(\theta^c, \theta^{s_1}, ..., \theta^{s_t}) = \min_{\substack{\theta^c \\ \theta^{s_1},...,\theta^{s_t}}} \begin{bmatrix} \mathcal{L}_1(\theta^c, \theta^{s_1}) \\ \mathcal{L}_2(\theta^c, \theta^{s_2}) \\ \vdots \\ \mathcal{L}_n(\theta^c, \theta^{s_t}) \end{bmatrix}^{\mathsf{T}} \tag{3.1}$$

where $\mathbf{L}$ is the full objective vector, $\mathcal{L}_1$, ..., $\mathcal{L}_t$ are t different objectives, respectively. θ^c is the common parameters shared by all objectives, while θ^{s_1}, ..., θ^{s_t} are the objective-specific parameters.

Notice that one of the key characteristics of a MOO problem is that a solution that

can optimize each objective to an ideal situation may not exist. This is exactly due to the conflict and correlation among the objectives as discussed before. The optimal solution of a MOO problem should balance all the objectives, which is called Pareto optimality.

Definition 1. Pareto Optimality

1. A solution θ_1 **dominates** another solution θ_2 if for all objectives $L_i(\theta_1^c, \theta_1^{s_i}) \leq L_i(\theta_2^c, \theta_2^{s_i})$, where $i \in \{1, ..., t\}$. Then there exists at least one objective $j \in \{1, ..., t\}$, where $L_j(\theta_1^c, \theta_1^{s_i}) < L_j(\theta_2^c, \theta_2^{s_i})$.

2. A solution is of **Pareto optimality** if there does not exist any other solution that dominates it.

3. There is usually more than one solution reaching Pareto optimality in a MOO problem. The set of such solutions is called **Pareto set**, which is the solution set of a MOO problem. The curve of the points in the **Pareto set** is called the **Pareto frontier**.

3.2 Multiple Gradient Descent Algorithm

Borrowing the idea from gradient descent on a single objective, the Multiple Gradient Descent Algorithm (MGDA) can be regarded as an extension of the gradient-based algorithm on multiple objectives. The overall objective of solving a MOO problem by

MGDA is usually a weighted summation of t single objectives, defined as:

$$\mathcal{L}(\theta) = \mathcal{L}(\theta^c, \theta^{s_1}, ..., \theta^{s_t}) = \sum_{i=1}^{t} \alpha_i \cdot \mathcal{L}_i(\theta^c, \theta^{s_i}) \,, \tag{3.2}$$

where the coefficients of all the objectives satisfy $\sum_{i=1}^{t} \alpha_i = 1$ and $\alpha_i \geq 0$, for $i = 1, ..., t$.

Before diving into the detail of MGDA, we need to be aware of what properties the Pareto optimal solution should have. Notice that there are no direct conditions for Pareto optimality; therefore, we introduce the Pareto stationary, which is a necessary condition for Pareto optimality in a MOO problem: a Pareto optimal solution must be Pareto stationary, while the reverse may not hold.

Definition 2. Pareto Stationarity

A solution θ^* is of **Pareto stationarity** if it satisfies all following conditions:

1. $\sum_{i=1}^{t} \alpha_i = 1$, $\alpha_i \geq 0$, for $i = 1, ..., t$,
2. $\sum_{i=1}^{t} \alpha_i \nabla_{\theta^{*c}} \mathcal{L}_i(\theta^{*c}, \theta^{*s_i}) = 0$,
3. $\nabla_{\theta^{*s_i}} \mathcal{L}_i(\theta^{*c}, \theta^{*s_i}) = 0$, for $i = 1, ..., t$.

The above conditions are also known as the **Karush-Kuhn-Tucker (KKT) conditions** first published in [Kuhn and Tucker, 1951].

Based on these, MGDA leverages Karush–Kuhn–Tucker (KKT) conditions to solve the MOO problem, which are necessary for Pareto optimal solutions. [Sener and Koltun,

2018] proposed to solve a quadratic-form constrained minimization problem defined as follows:

$$\begin{aligned} \min_{\alpha_1,\alpha_2,...,\alpha_t} & \left\| \sum_{i=1}^{t} \alpha_i \cdot \nabla_{\theta^c} \mathcal{L}_i(\theta^c, \theta^{s_i}) \right\|_2^2 , \\ \text{s.t.}, & \sum_{i=1}^{t} \alpha_i = 1, \alpha_i \geq 0, \text{for} \quad i = 1, ..., t \, . \end{aligned} \tag{3.3}$$

Given Eq. 3.3, there are two situations for the final solution: the final solution is Pareto stationary if the solution to this optimization problem is 0; otherwise, the solution offers a common descent direction which benefits all the objectives as proved by [Désidéri, 2012]. Therefore, one can use the single-objective gradient descent for optimizing the objective-specific parameters θ^{s_i} on t different objectives and employ the obtained solution to the above equations for updating the common parameters θ^c.

3.3 Solving the MOO Problem

We first introduce a special case where there are only two objectives in the loss function:

$$\min_{\alpha \in [0,1]} \|\alpha \cdot \nabla_{\theta^c} \mathcal{L}_1(\theta^c, \theta^{s_1}) + (1-\alpha) \cdot \nabla_{\theta^c} \mathcal{L}_1(\theta^c, \theta^{s_2})\|_2^2 \, . \tag{3.4}$$

The analytical solution to this quadratic problem is:

$$\alpha^* = \frac{(\nabla_{\theta^c} \mathcal{L}_2(\theta^c, \theta^{s_2}) - \nabla_{\theta^c} \mathcal{L}_1(\theta^c, \theta^{s_1}))^\intercal \nabla_{\theta^c} \mathcal{L}_2(\theta^c, \theta^{s_2})}{\|\nabla_{\theta^c} \mathcal{L}_1(\theta^c, \theta^{s_1}) - \nabla_{\theta^c} \mathcal{L}_1(\theta^c, \theta^{s_2})\|_2^2}, \tag{3.5}$$

Algorithm 1: Frank-Wolf Solver [Frank and Wolfe, 1956, Sener and Koltun, 2018]

Result: A list of learned scaling coefficients: α_1, ..., α_t
$t \leftarrow$ number of objectives
$\theta \leftarrow$ model parameters: $(\theta^c, \theta^{s_1}, ..., \theta^{s_t})$
Initialization: $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_t) = (\frac{1}{t}, ..., \frac{1}{t})$
Precompute M, $\forall i, j \in \{1, ..., t\}$:
$M_{ij} = (\nabla_{\theta^c} \mathcal{L}_i(\theta^c, \theta^{s_i}))^\intercal (\nabla_{\theta^c} \mathcal{L}_j(\theta^c, \theta^{s_j}))$
repeat
$i^* = \text{argmin}_r \sum_i \alpha_i M_{ri}$
$w^* = \text{argmin}_w ((1-w)\boldsymbol{\alpha} + w\boldsymbol{e}_{i^*})^\intercal M ((1-w)\boldsymbol{\alpha} + w\boldsymbol{e}_{i^*})$
$\boldsymbol{\alpha} = (1 - i^*)\boldsymbol{\alpha} + w^* \boldsymbol{e}_{i^*}$
until w^* *converge*;
return $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_t)$

where the α^* should be clipped into $[0, 1]$.

Although there are no analytical solutions for more than two objectives in a MOO problem, we can still utilize the analytical solution of two objectives to conduct the line search efficiently. This technique is proposed in [Sener and Koltun, 2018] based on the Frank-Wolf algorithm [Frank and Wolfe, 1956, Jaggi, 2013], where the details are shown in Algorithm 1.

3.4 Solution Selection

There is no consensus strategy on choosing a Pareto optimal solution from a Pareto set since there is not always a solution strictly dominating all others. To select a proper solution, we borrow the idea from one of the most well-known metrics in Theoretical Economics, the *Least Misery Strategy* [Pessemier et al., 2014], for guiding us to select a "fair" solution for all the objectives.

Algorithm 2: Multi-FR Framework

```
Initialization( )
for i ∈ 1, ..., t do
    Construct individual objective L_i(Θ)
end
for epoch ∈ 1, ..., n_epoch do
    for batch ∈ 1, ..., n_batch do
        Forward_Passing( )
        for i ∈ 1, ..., t do
            Compute gradient for each objective: ∇_Θ L_i(Θ)
            Gradient_Normalization( )  (optional)
        end
        α = (α_1, ..., α_t) ← Frank-Wolf Solver(t, Θ)
        Construct single aggregated objective: L(Θ) = Σ_{i=1}^{t} α_i · L_i(Θ)
        ∇_Θ L(Θ) = Σ_{i=1}^{t} α_i · ∇_Θ L_i(Θ)
        Θ updation
    end
end
```

Motivated by the *Least Misery Strategy*, our recommendation aims to minimize the highest loss function of the objectives:

$$\min \max\{\mathcal{L}^1, \mathcal{L}^2, ..., \mathcal{L}^q\}, \tag{3.6}$$

where $\mathcal{L}^i$ is the aggregated single objective after finishing the i^{th} round of our model, and q is the total number of rounds running the model. Therefore, given a generated Pareto frontier by running Algorithm 1 and Algorithm 2 for multiple rounds (q rounds) in our proposed model, the final recommendation is the solution with the minimum value of Eq. 3.6.

Chapter 4

Objective Construction

In this section, we present the objectives applied in the *Multi-FR*, which satisfy the needs of both satisfaction and fairness in the top-k recommendation on both the consumer and producer side.

4.1 User-Item Interaction Modeling

Since the proposed *Multi-FR* is employed in the recommendation scenario, the first objective is to measure recommendation quality. We treat all the training data as user implicit feedback and optimize the proposed framework by the Bayesian Personalized Ranking (BPR) objective [Rendle et al., 2012]: optimizing the pairwise ranking between

the positive and non-observed items:

$$\mathcal{L}_{ranking} = \underset{\Theta}{\operatorname{argmin}} \sum_{u,i,j \in D_s} -\log \sigma(\hat{x}_{uij}) + \lambda \left\|\Theta\right\|_2^2, \tag{4.1}$$

where $\Theta = [\![\Theta^U, \Theta^I]\!]$ is the model parameter containing user embeddings and item embeddings, D_s is the constructed training set, i denotes the positive item in the training set and j denotes the randomly sampled negative item, $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$ denotes the user u's preference of item i over item j. We use the inner product to calculate the relevance score $\hat{x}_{ui}$ between u and i as: $\hat{x}_{ui} = \langle \Theta_u^U, \Theta_i^I \rangle$, where $\langle , \rangle$ denotes the inner product between two vectors.

4.2 Fairness-aware Recommendation

Fairness is a big concern in current information retrieval systems, which has a huge impact on the multi-stakeholder marketplaces. In our proposed *Multi-FR*, we consider the group fairness on both the consumer (customer) and producer (item) sides.

4.2.1 Fairness Constraints on the Consumer Side

It has been shown that sensitive features influence the satisfaction of consumers in recommendation [Zhu et al., 2018]. For instance, the sensitivity of gender attributes in a job recommendation marketplace is recognized and analyzed in [Singh and Joachims,

2018]. Regarding the group fairness of consumers, we want the satisfaction of different groups with sensitive features to be almost the same. Considering there are n groups among the consumer side, the group fairness/disparity can be defined as the difference between mean satisfaction values:

$$\mathcal{L}_{GFair\_c} = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n} \sum_{j=1}^{i} \|\mathbf{s}_i - \mathbf{s}_j\|_2^2 \ , \tag{4.2}$$

where $\binom{n}{2}$ is to compute the number of combination for the pair-wise comparisons, $\mathbf{s}_i$ is the average satisfaction representation of users in i^{th} group. Normally, ranking-based metrics are used to measure users' satisfaction concerning the recommendation quality. Here we adopt Normalized Discounted Cumulative Gain at K (NDCG@K), a widely-used ranking metric, to measure the satisfaction $\mathbf{s}$ of users.

However, NDCG@K only measures the recommendation quality at rank K. Solely considering the recommendation equality at rank K does not necessarily guarantee the results are also fair at rank $K-1, K-2, ..., 1$. Thus, we construct the $\mathbf{s}_i \in \mathbb{R}^K$ as an NDCG vector for the i^{th} group among the consumers, where each entry represents a value NDCG@k ($k = 1, ..., K$). Specifically,

$$\mathbf{s}_i = \frac{\mathbf{G} \cdot \mathbf{m}_i}{b_i} \tag{4.3}$$

where $\mathbf{G} \in \mathbb{R}^{b \times K}$ is a matrix containing NDCG@1 to NDCG@K for all users in one batch

size, $\mathbf{m}_i \in \mathbb{R}^b$ represents the mask of the i^{th} group (1 indicates "belong", 0 otherwise). b_i refers to the number of consumers who belong to the i^{th} group in one batch.

Any kinds of attributes can be adopted for computing the mask $\mathbf{m}_i$ in Eq. 4.3. In our model, we consider two types of disparities regarding consumers' two most common and sensitive attributes.

Gender-based Fairness. Gender is one of the most sensitive attributes of humans and many works have already presented insightful observations and analysis on gender bias in Internet services [Kay et al., 2015, Butterly, 2015].

We construct the gender mask $\mathbf{m}_f$ and $\mathbf{m}_m$ for females and males and aim to minimize the satisfaction difference between these two groups. Note that gender is treated as a binary class due to the available labels in the dataset. We do not intend to suggest that gender identities are binary, nor support any such assertions.

Age-based Fairness. Other than gender, we also consider age-based fairness. We construct $\mathbf{m}_{a_i}$ as the mask for the i-th age group. We split the age into 7 stages following the criterion in the MovieLens datasets [Harper and Konstan, 2016]. Then the age-based fairness constraint is to minimize the difference of the satisfaction vectors among all age groups, as described in Eq. 4.2.

4.2.2 Fairness Constraints on the Producer Side

Previous works mainly focus on the fair satisfaction of the consumer side,[Bose and Hamilton, 2019], which implicitly assumes that the users are the only stakeholder in a recommendation system. However, the fairness of recommended items should also be considered since it represents the benefits of the producers, which is an even more significant stakeholder in a commerce marketplace. This problem has been noticed in the community recently [Mehrotra et al., 2018, Patro et al., 2020].

As for the group fairness on producers, the goal is to find a ranking strategy that can offer a fair probability of exposure on items based on their merits. However, one of the key challenges, as mentioned in [Diaz et al., 2020], is that a single fixed ranking for a query (in retrieval) or context (in recommendation) tend to limit the ability of an algorithm to distribute exposure amongst relevant items. For a static ranking, (i) some relevant items may receive more exposure than other relevant items, and (ii) some irrelevant items may receive more exposure than other relevant items. Therefore, we hope to find a policy that samples a permutation from a distribution over the set of all permutations of $|I|$ items, and such a stochastic ranking policy will be able to force all items to receive a fair exposure proportional to their merits, thus achieving fair expected exposures.

Assume we cluster the items (producers) into z groups, then the fairness on the

producer side can be defined as the difference of the two exposure distributions:

$$\mathcal{L}_{GFair\_p} = \|\epsilon - \epsilon^*\|_2^2 \,, \tag{4.4}$$

where $\epsilon \in \mathbb{R}^z$ is a vector representing the distribution of exposure on items from different groups, $\epsilon^* \in \mathbb{R}^z$ is the target exposure distribution vector, which is generally defined as a flat distribution: $\epsilon^* = [\frac{1}{z}, ..., \frac{1}{z}]$. One can freely define it as the corpus distribution or any specific distribution under special circumstances.

Consider a query of items, the relevance score of the i^{th} item is l_i to this query. We can obtain the sampling probability of an item by using a Plackett-Luce model [Plackett, 1975], and we refer to this as the Plackett-Luce (PL) policy:

$$p_i = \frac{\exp(l_i)}{\sum_{j \in I} \exp(l_j)} \,. \tag{4.5}$$

thus construct a ranking by sampling items sequentially.

Assume the batch size is b and the number of picked relevant items of each user is n_r, then we can obtain a matrix $R \in \mathbb{R}^{b \times n_r}$ containing the ranks of all these relevant items on b queries. To calculate the exposure, we adopt the position-biased assumption that a user's probability of visiting a position decreases exponentially with rank [Diaz et al., 2020, Moffat and Zobel, 2008]. Then we can compute the exposure of all relevant items in a batch as $E = \gamma^R$, where γ represents the patience parameter and controls how deep the

user is likely to browse in a ranking. Then, the group-wise item exposure ϵ is computed as:

$$\epsilon = E \cdot \mathbf{M}_g \,, \tag{4.6}$$

where $\mathbf{M}_g$ is the group mask.

We consider two types of group fairness constraints on the producer side. Both are based on the same framework defined in Eq. 4.4, the only difference lies in the different construction of the group mask in the preprocessing.

Popularity-based Fairness. The "superstar economics" [Mehrotra et al., 2018, Baranchuk et al., 2011] always occurs in real-world recommendation scenarios, where a small number of most popular artists possessed most of the exposures from customers. This leads to the lock-in of popular products and items, especially for users who want to minimize access costs. A major side-effect of superstar economics is the impedance to suppliers on the tail-end of the spectrum, who struggle to attract consumers and are not satisfied with the marketplace.

To construct the popularity-based group mask $\mathbf{M}_{g-pop}$, we rank all the $|I|$ items based on their occurrences in the dataset from the highest to the lowest and evenly split them into 5 groups labeled from 5 to 1, where each group contains 20% of items. Then we define $\mathbf{M}_{g-pop} \in \mathbb{R}^{5\times|I|}$, where each entry (i, j) in the matrix is 1 if the j^{th} item belongs to the i^{th} popularity group, and 0 otherwise.

Genre-based Fairness. In some specific scenarios, like movie recommendations, genre-based fairness is also worthy of considering. Therefore, we propose a disparity measure on the producer side based on the genre of items. We make use of all the genres in the MovieLens datasets [Harper and Konstan, 2016], and build a binary matrix mask of all genres on all movies. We adopt the same strategy as described in the popularity-based fairness, and we get the mask $\mathbf{M}_{g-genre} \in \mathbb{R}^{g \times |I|}$, where g is the number of movie genres.

4.2.3 Differentiable Approximation of the Ranking

In our formulation, relevance is defined as a function of the ranked list of items, but the sorting operation is inherently non-differentiable. To mitigate this problem, we adopt the continuous approximation of the ranking function proposed in [Wu et al., 2009, Qin et al., 2010] that is amenable to gradient-based optimization. The key insight behind these approximations lies in defining the rank of an item in terms of the pairwise preference with every other item in the collection:

$$r_i = 0.5 + \sum_{j}^{n} \sigma'(s_j - s_i), \text{ where } \sigma'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0.5, & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{4.7}$$

The discrete function $\sigma(\cdot)$ is typically approximated using the differentiable sigmoid function.

Given the approximated differentiable ranks of items, it is then straightforward to derive an optimization objective for standard relevance metrics—*e.g.*, discounted cumulative gain (DCG)—that can be directly optimized using gradient descent:

$$\text{SmoothDCG} = \sum_{i}^{n} \frac{\text{rel}_i}{\log_2(r_i + 1)} \,. \tag{4.8}$$

Therefore, we adopt such SmoothDCG when constructing the fairness objective in Eq. 4.3 on the consumer side during training, but still, use the original NDCG in the evaluation phase.

As for the producer side, in order to mitigate the same non-differentiable operation in Eq. 4.5, we adopt the Gumbel Softmax technique proposed in [Maddison et al., 2016, Bruch et al., 2020]: we reparameterize the probability distribution by adding independently-drawn noise ζ sampled from the Gumbel distribution to l and sorting items by the "noisy" probability distribution $\tilde{p}_i$:

$$\tilde{p}_i = \frac{\exp(l_i + \zeta_i)}{\sum_{j \in I} \exp(l_j + \zeta_j)} \,. \tag{4.9}$$

After obtaining the perturbed probability distribution $\tilde{p}_i$, we then compute the smooth

rank [Wu et al., 2009] of each item as:

$$r_i = \sum_{j \in I, j \neq i} \left(1 + \exp\left(\frac{\tilde{p}_i - \tilde{p}_j}{\tau}\right)\right)^{-1}, \tag{4.10}$$

where the temperature τ is a hyperparameter and controls the smoothness of the approximated ranks. Then the exposure E in Eq. 4.6 is computed based on the smooth ranking; thus the fairness objective is differentiable during the training procedure.

4.3 Overall Training Objective

Our overall training objective is a weighted summation of the ranking loss and the disparity loss:

$$\begin{aligned} \mathcal{L} &= \alpha \cdot \mathcal{L}_{ranking} + \sum_{i=1}^{m} \beta^i \cdot \mathcal{L}^i_{GFair\_c} + \sum_{i=1}^{n} \gamma^i \cdot \mathcal{L}^i_{GFair\_p}, \\ \text{s.t. }, \alpha &+ \sum_{i=1}^{m} \beta^i + \sum_{i=1}^{n} \gamma^i = 1, \alpha \geq 0, \beta^i \geq 0, \gamma^i \geq 0. \end{aligned} \tag{4.11}$$

Here, m and n refer to the number of fairness constraints on the consumer and producer sides, respectively.

It is worth noticing that our proposed *Multi-FR* framework does not rely on specific formulations of the loss functions or the model structures. Although the four aforementioned disparity measures all belong to the group fairness, one can also define any individual fairness objectives and scalably apply them into our framework as long

as the gradient is available.

Chapter 5

Experiments and Evaluation

In this chapter, we evaluate the proposed model and other baseline methods on three real-world datasets.

5.1 Datasets

The proposed model is evaluated on three real-world datasets from various domains with different sparsities: *MovieLens100K*, *MovieLens1M* [Harper and Konstan, 2016], and *last.fm* (lastfm360k). *MovieLens100K* and *MovieLens1M* are user-movie datasets collected from the *MovieLens* website. These two datasets provide 100 thousand and 1 million user-movie interactions, respectively, and the user metadata (gender and age group) and movie genres. The *last.fm* data is collected from the Last.fm website, which contains the music listening records of 360 thousand users along with the gender and age of users.

	ML100k	ML1M	last.fm
#User	943	5,805	19,234
#female/#male/#age	273/670/7	1,655/4,150/7	4,022/15,212/7
#Item	1,682	3,574	9,703
#genre/#popularity	18/5	18/5	-/5
#Interaction	100,458	678,740	1,049,322
Density	6.33%	3.27%	0.56%

Table 5.1: The statistics of datasets.

Under the implicit feedback setting, we keep those ratings no less than four (out of five) as positive feedback and treat all other ratings as missing entries for all datasets. To filter noisy data, we only keep the users with at least ten ratings and the items at least with five ratings. We adopt the age group strategy of the MovieLens dataset to split users into 7 different age groups and the movies into 18 different genres in all experiments. For all the datasets, we also group the items into 5 different groups based on their popularity. For each user, we randomly select 20% of the rated items as ground truth for testing, The remaining 70% and 10% data constitutes the training and validation set. The data statistics after preprocessing are shown in Table 5.1.

5.2 Evaluation Metrics

In this section, we demonstrate our chosen metrics on recommendation accuracy, fairness, and diversity. We adopt both self-defined metrics and commonly used measurement in academia. Our measurement of fairness and diversity covers the *individual level*, *group level*, and *system level*.

Metrics for measuring recommendation accuracy:

- **Recall@k**, which indicates the percentage of her rated items that appear in the top k recommended items.
- **NDCG@k**, which is the normalized discounted cumulative gain at k, which takes the position of correctly recommended items into account.

Self-defined Metrics for measuring fairness:

- **Disparity**$_u$ measures the unfairness on the user side, i.e. Eq. 4.2.
- **Disparity**$_i$ measures the unfairness on the item side, i.e. Eq. 4.4.

General metrics for measuring fairness and diversity:

- **Gini Index** measures the inequality among values of a frequency distribution [Gin, 2008], e.g., numbers of occurrences (exposures) in the recommendation list. This measurement is at *individual level*. Given a list of exposure of all items (I) in the recommendation list, $l = [e_1, e_2, ..., e_{|I|}]$, the Gini Index is calculated as:

$$\text{Gini}(l) = \frac{1}{2|I|^2\overline{e}} \sum_{i=1}^{|I|} \sum_{j=1}^{|I|} |e_i - e_j|, \tag{5.1}$$

  where $\overline{e}$ is the mean of all item exposures.

- **Popularity rate** computes the proportion of popular items in the recommendation list

against the total number of items in the list, which can be regarded as a *group level* measurement.

- **Simpson's diversity index**, which is a *system-level* measurement of diversity which takes into account the number of species present, as well as the relative abundance of each specie [Sim, 1949]. Given a list of exposures of all items in the recommendation results and the group label of each, the Simpson's diversity index can be formulated as:

$$\text{Diversity} = 1 - \left(\frac{\sum_{i=1}^{g} n_i(n_i - 1)}{N(N-1)}\right), \tag{5.2}$$

where g is the total number of groups, n_i is the total number of items of group i, and N is the total number of items of all groups.

5.3 Method Studied

We choose three models as our base models:

- **BPRMF**, Bayesian Personalized Ranking-based Matrix Factorization [Rendle et al., 2009], which is a classic method for learning pairwise personalized rankings from user implicit feedback.

- **WRMF**, Weighted Regularized Matrix Factorization [Hu et al., 2008], which minimizes the square error loss by assigning both observed and unobserved feedback

with different confidential values based on matrix factorization.

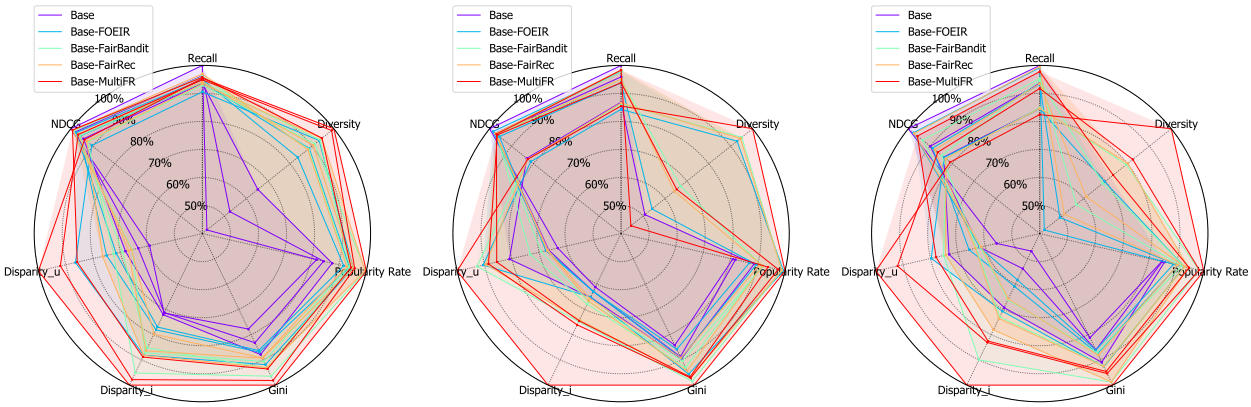
- **NGCF**, Neural Graph Collaborative Filtering [Wang et al., 2019]. This method integrates the user-item interactions into the embedding learning process, and exploits the graph structure by propagating embeddings on it to model the high-order connectivity.

We first use the above three base models to learn the latent representation of users and items and obtain the relevance scores between them. Then we adopt the following three fairness-aware approaches on the top of the three base models to achieve fair recommendation for a comparison with our model.

- **FOEIR**, Fairness of Exposure in Rankings [Singh and Joachims, 2018], which is a fairness-aware algorithm incorporating a standard linear program and the Birkhoff-von Neumann decomposition [Birkhoff, 1967].

- **FairBandit**, which is a fairness-aware method formalizing the recommendation problem as a combinatorial contextual bandit problem, wherein the recommendation system powering the two-sided marketplace repeatedly interacts with consumers. It aims to balance the trade-off between the relevance of recommendations to the consumer and fairness of representation of suppliers [Mehrotra et al., 2018].

- **FairRec**, which is a two-sided fairness-aware method achieving envy-freeness up-to-one on the user side and exposure guarantee on the item side [Patro et al., 2020]. It is



(a) MovieLens 100K (b) MovieLens 1M (c) last.fm

Figure 5.1: Relative performance achievement comparing to the best overall performance for each *Metric@20* on three datasets. We use the best performance value in each column as the numerator for *NDCG*, Recall, and *Diversity* (the larger, the better), while as the denominator for the others (the smaller, the better). Each model setting has three lines since the *Base* may refer to BPRMF, WRMF, or NGCF. We colour the area besieged by the best performance points on each metric for each model setting.

motivated by the fair allocation [Bouveret et al., 2016] and adopts the Greedy-Round-Robin algorithm [Biswas and Barman, 2018, Caragiannis et al., 2019] to allocate item candidates to users.

Lastly, we adopt our proposed **Multi-FR** method on the top of the BPRMF, WRMF, and NGCF to form our final model. Our method allows the weights on different objectives to be adaptively learned during the training process with the model embeddings.

Model	Recall ↑		NDCG ↑		Disparity_u ↓		Disparity_i ↓		Gini ↓		Popularity rate ↓		Diversity ↑	
	k=10	k=20	k=10	k=20	k=10	k=20	k=10	k=20	k=10	k=20	k=10	k=20	k=10	k=20
MovieLens-100k														
BPRMF	0.2152	0.3210	0.2637	0.2848	1.3580	2.8215	1.2131	1.2237	0.6919	0.6840	0.8996	0.8630	0.1838	0.2367
WRMF	0.2166	0.3305	0.2748	0.2964	1.3753	3.0307	1.2215	1.2382	0.7278	0.7223	0.9256	0.8953	0.1391	0.1905
NGCF	**0.2275**	**0.3431**	**0.2855**	**0.3022**	1.4226	3.2431	1.2333	1.2347	0.7526	0.7728	0.9621	0.9233	0.1023	0.1525
BPRMF-FOEIR	0.1968	0.3107	0.2473	0.2734	1.2103	2.2250	1.0023	1.0011	0.6558	0.6540	0.8351	0.7934	0.2817	0.3381
WRMF-FOEIR	0.2107	0.3221	0.2694	0.2896	1.2533	2.5632	1.1185	1.1296	0.7152	0.6916	0.8574	0.8051	0.2478	0.3217
NGCF-FOEIR	0.2242	0.3259	0.2705	0.2953	1.2624	2.6877	1.1388	1.1465	0.7239	0.6966	0.8627	0.8234	0.2282	0.3029
BPRMF-FairBandit	0.2070	0.3189	0.2511	0.2792	1.0923	2.6527	0.9353	0.9270	0.6251	0.6230	0.8199	0.7572	0.2819	0.3310
WRMF-FairBandit	0.2133	0.3225	0.2681	0.2900	1.2638	2.7769	0.9734	1.0352	0.6734	0.6444	0.8229	0.8034	0.2625	0.3394
NGCF-FairBandit	0.2253	0.3297	0.2732	0.2983	1.3236	2.9254	1.0039	1.0125	0.6925	0.6632	0.8321	0.8089	0.2622	0.3205
BPRMF-FairRec	0.2049	0.3204	0.2495	0.2801	1.3627	2.5129	0.9357	1.0001	0.6325	0.6459	0.8214	0.7589	0.2916	0.3426
WRMF-FairRec	0.2140	0.3227	0.2735	0.2925	1.3624	2.8691	0.9626	1.0214	0.6954	0.6729	0.8315	0.7926	0.2621	0.3281
NGCF-FairRec	0.2252	0.3327	0.2776	0.2996	1.3526	2.9162	1.0221	1.1056	0.7027	0.6735	0.8410	0.8134	0.2518	0.3194
BPRMF-MultiFR	0.2055	0.3273	0.2516	0.2833	**0.9877**	**1.9234**	**0.8235**	**0.8826**	**0.6027**	**0.6011**	0.8032	**0.7552**	0.3029	**0.3625**
WRMF-MultiFR	0.2156	0.3255	0.2733	0.2921	0.9997	2.0913	0.8672	0.9023	0.6239	0.6124	**0.8021**	0.7889	**0.3045**	0.3588
NGCF-MultiFR	0.2245	0.3286	0.2752	0.3000	1.0232	2.2421	0.9862	0.9928	0.6428	0.6421	0.8213	0.8001	0.3042	0.3429
MovieLens-1M														
BPRMF	0.1462	0.2287	0.2360	0.2438	1.5225	3.2123	1.2648	1.2638	0.7586	0.7512	0.9326	0.9047	0.1264	0.1743
WRMF	0.1681	0.2525	0.2850	0.2859	2.1773	3.9079	1.3125	1.3105	0.7720	0.7579	0.9921	0.9808	0.0157	0.0377
NGCF	**0.1782**	**0.2633**	**0.2852**	**0.2936**	2.5632	4.1124	1.3527	1.3469	0.8010	0.7992	0.9935	0.9922	0.0032	0.0123
BPRMF-FOEIR	0.1425	0.2220	0.2318	0.2384	1.4263	2.8707	1.2624	1.2660	0.7171	0.7034	0.8530	0.8075	0.2545	0.3183
WRMF-FOEIR	0.1646	0.2469	0.2809	0.2804	2.1750	3.8470	1.3128	1.3105	0.7710	0.7534	0.9221	0.8967	0.0157	0.1854
NGCF-FOEIR	0.1762	0.2574	0.2834	0.2890	2.3345	3.8728	1.3189	1.3098	0.7786	0.7842	0.9305	0.9231	0.0127	0.1026
BPRMF-FairBandit	0.1434	0.2272	0.2339	0.2425	1.3925	2.8013	0.9728	1.1008	0.6877	0.6831	0.8439	0.8122	0.2598	0.3257
WRMF-FairBandit	0.1652	0.2501	0.2842	0.2840	1.8729	3.5467	1.0927	1.2129	0.7230	0.7234	0.9000	0.8762	0.1001	0.2301
NGCF-FairBandit	0.1752	0.2575	0.2839	0.2851	2.3014	3.7529	1.2438	1.2320	0.7229	0.7439	0.9024	0.8729	0.1002	0.1252
BPRMF-FairRec	0.1453	0.2280	0.2344	0.2425	1.4927	2.9012	1.0826	1.1109	0.6927	0.6931	0.8531	0.8123	0.2637	0.3237
WRMF-FairRec	0.1661	0.2502	0.2846	0.2851	2.1023	3.8721	1.1352	1.2358	0.7241	0.7129	0.9027	0.8749	0.1015	0.2203
NGCF-FairRec	0.1774	0.2591	0.2848	0.2856	2.5413	3.8927	1.2635	1.2533	0.7309	0.7542	0.9135	0.8862	0.1000	0.1224
BPRMF-MultiFR	0.1458	0.2252	0.2333	0.2424	**1.0235**	**2.5972**	**0.8716**	**0.8241**	**0.6825**	**0.6728**	**0.8214**	**0.8027**	**0.3023**	**0.3426**
WRMF-MultiFR	0.1644	0.2470	0.2811	0.2832	1.6523	2.9341	0.9726	1.0826	0.7032	0.6923	0.8527	0.8231	0.1029	0.2239
NGCF-MultiFR	0.1724	0.2588	0.2829	0.2844	1.8528	3.0375	1.1125	1.1057	0.7152	0.6955	0.8734	0.8562	0.0965	0.1524
Last.fm														
BPRMF	0.1245	0.1904	0.1669	0.1892	1.3277	1.3658	1.3099	1.3103	0.8136	0.8161	0.9893	0.9792	0.0211	0.0407
WRMF	0.1322	0.2104	0.1826	0.2031	1.6231	1.6127	1.5135	1.6852	0.8523	0.8627	0.9905	0.9889	0.0104	0.0214
NGCF	**0.1452**	**0.2247**	**0.1923**	**0.2258**	1.8231	1.7923	1.8326	1.9349	0.9006	0.9138	0.9932	0.9905	0.0096	0.0102
BPRMF-FOEIR	0.1245	0.1899	0.1669	0.1888	1.2746	1.2541	1.2801	1.2837	0.8136	0.8006	0.9893	0.9033	0.0211	0.1752
WRMF-FOEIR	0.1322	0.2096	0.1825	0.2008	1.5268	1.5179	1.3687	1.4920	0.8523	0.8489	0.9906	0.9258	0.0104	0.1237
NGCF-FOEIR	0.1428	0.2229	0.1899	0.2206	1.6092	1.6138	1.5247	1.5562	0.9006	0.8623	0.9932	0.9429	0.0096	0.1058
BPRMF-FairBandit	0.1233	0.1908	0.1661	0.1870	1.1212	1.3298	1.0987	1.0091	0.7774	0.7515	0.9433	0.8536	0.1125	0.1998
WRMF-FairBandit	0.1332	0.2125	0.1810	0.1913	1.4732	1.5823	1.2110	1.2452	0.8028	0.8035	0.9527	0.9015	0.1124	0.1729
NGCF-FairBandit	0.1429	0.2142	0.1897	0.2111	1.5110	1.6176	1.3724	1.3722	0.8433	0.8521	0.9813	0.9126	0.1024	0.1425
BPRMF-FairRec	0.1240	0.1902	0.1659	0.1872	1.3320	1.3435	1.1511	1.1627	0.7826	0.7519	0.9625	0.8892	0.1027	0.2016
WRMF-FairRec	0.1322	0.2100	0.1820	0.1984	1.5826	1.5726	1.2231	1.2338	0.8038	0.8023	0.9699	0.9022	0.1008	0.1539
NGCF-FairRec	0.1435	0.2236	0.1901	0.2197	1.6235	1.6282	1.3825	1.3791	0.8522	0.8425	0.9826	0.9273	0.0927	0.1286
BPRMF-MultiFR	0.1209	0.1853	0.1592	0.1726	**1.0288**	**0.9999**	**0.9323**	**0.9083**	**0.7514**	**0.7426**	**0.9023**	**0.8388**	**0.1674**	**0.2515**
WRMF-MultiFR	0.1301	0.2062	0.1784	0.1954	1.1273	1.0862	1.0824	1.0927	0.7823	0.7782	0.9275	0.8526	0.1462	0.2073
NGCF-MultiFR	0.1426	0.2197	0.1888	0.2164	1.2526	1.2830	1.1081	1.1001	0.8002	0.7849	0.9388	0.8862	0.1388	0.1848

Table 5.2: Summary of the performance. We evaluate for accuracy (*Recall* and *NDCG*) and fairness (Disparity_u, Disparity_i, *Gini*, *Popularity rate*, and *Diversity*), where k is the length of the recommendation list. A metric followed by "↑" means "the larger, the better", while a metric followed by "↓" means "the smaller, the better". All results are significant at $p < 0.01$.

5.4 Experiment Settings

In the experiments, we optimize all models using the Adam optimizer with the Xavier initialization [Glorot and Bengio, 2010]. The embedding size is fixed to 50 and the batch size to 1024 for all baseline models. The learning rate and the regularization hyper-parameter are selected from $\{1e^{-1}, 1e^{-2}, 1e^{-3}, 1e^{-4}, 1e^{-5}\}$. The patience parameter γ is selected from $\{0.25, 0.5, 0.75\}$. The smooth temperature in SmoothRank is selected from $\{1e^{-4}, 1e^{-5}, 1e^{-6}\}$. The K value in NDCG@K used for computing the consumer-side fairness described in Section 4.2.1 is selected from $\{20, 50, 100\}$. For all the datasets, we randomly sample one unobserved item as the negative sample for each user to speed up the training process. Further, for the FOEIR model, since it requires to solve a linear program with size $|I| \times |I|$ for each consumer with huge computational costs, we rerank the top-100 items from the base model then select the new top-K (K<100) as the final recommendation. For the FairBandit approach, we adopt the interpolated recommendation policy as in the original paper with a scaling factor $\beta = 0.5$. Early stopping strategy is performed, i.e., permutate stopping if *Recall@20* on the validation data does not increase for 50 successive evaluation steps, for which the evaluation process is conducted for every five epochs. All experiments are conducted with PyTorch running on GPU machines (Nvidia Tesla P100).

5.5 Experimental Results and Analysis

5.5.1 Overall Performance Comparison

Table 5.2 summarizes all methods' best results on three datasets. Bold scores are the best in each column, while underlined scores are the second best.

Our model achieves obvious and significant improvements regarding all the fairness and diversity metrics. For instance, on the *ML100k* dataset, considering the top-10 recommendation, BPRMF-MultiFR reduces the disparity on the user side by 27.27% and 32.12% on the item side compared with the BPRMF base model. WRMF-MultiFR reduces the Gini index and Popularity rate by 13.04% and 13.34%, respectively. And NGCF-MultiFR model improves the system's diversity from 0.1391 to 0.3045, which is a rather great enhancement. The biggest improvement of the diversity metric is on the *last.fm* dataset, where the diversity measure is improved from 0.0211 to 0.1674 by BPRMF-MultiFR compared with the corresponding base model. WRMF-MultiFR and NGCF-MultiFR also largely enhance the diversity by a large margin. Furthermore, compared with other state-of-the-art fair ranking methods, Multi-FR can still consistently achieve better fairness measures on both sides.

We also observe a conflict between the recommendation accuracy and fairness. For instance, NGCF achieves the highest accuracy regarding Recall and NDCG on three datasets; however, its recommendation is the least fair and diverse compared to other

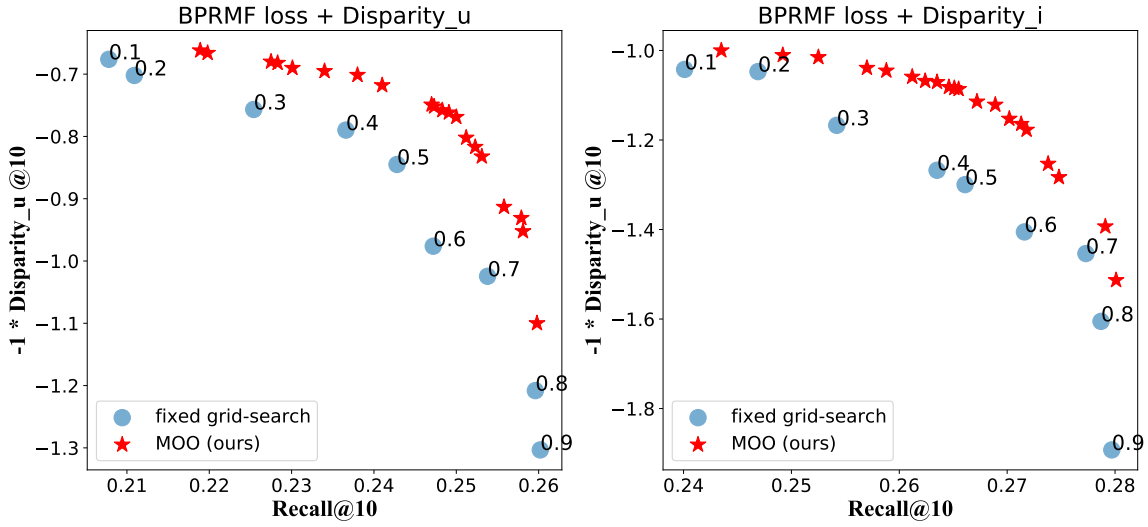


Figure 5.2: The Pareto frontier of our solutions versus the fixed grid-search solutions on the MovieLens 1M dataset.

models. FOEIR, FairBandit, and FairRec achieve better fairness by re-ranking the recommendation list based on the relevance scores obtained from the base models; however, the original ranking order is disrupted, leading to the accuracy drop. Our Multi-FR can balance the accuracy and fairness well by largely improving the fairness and diversity with little drop in the accuracy. For instance, concerning Recall@20, NGCF-MultiFR only has a drop of 4.23%, 1.71%, and 2.25% on three datasets, respectively, compared to the NGCF model. Considering the large magnitude of fairness and diversity improvements, we denote this accuracy drop is relatively small. This conclusion can also be obtained from Fig. 5.1.

Objective	*ML100K*	*ML1M*	*last.fm*
(1) BPRMF+1 D_u	676.4	11,059.2	69,438.3
(2) BPRMF+1 D_i	363.8	8,945.1	54,281.8
(3) BPRMF+2 D_u	749.3	15,044.0	98,177.2
(4) BPRMF+2 D_i	512.5	12,684.2	90,864.5
(5) BPRMF+1 D_u+1 D_i	912.7	19,560.7	102,232.1
(6) BPRMF+2 D_u+1 D_i	1,119.2	23,653.5	123,171.7
(7) BPRMF+1 D_u+2 D_i	1,013.8	23,189.3	120,816.9
(8) BPRMF+2 D_u+2 D_i	1,213.5	25,793.9	165,287.6

Table 5.3: The training efficiency comparison of different number of fairness constraints by using our model, *Multi-FR*. The training time is reported in seconds.

5.5.2 MOO vs Fixed Grid-search

In order to demonstrate the effectiveness of the MOO mechanism in *Multi-FR*, we conduct an experiment to compare our model with the grid-search strategy, where scaling factors on the BPRMF objective and fairness objective are manually set (the summation is 1). We only consider two-loss objectives for a convenient grid-search, which means we only add the disparity constraint on one side once training with the BPR ranking loss. The scatter plots are shown in Fig. 5.2. Each blue point indicates a grid-search solution averaged by 5 rounds where the value on the point is the weight on the BPR loss. Each red point refers to one final *Multi-FR* solution selected by the strategy described in Section 3.4 after 5 rounds. From the curve, we can observe that the MOO successfully balances the trade-off between fairness and recommendation quality. The clear margin distance between the curve of the red points (Pareto frontier) and the curve of the blue points show the effectiveness of the MOO mechanism in *Multi-FR*.

5.5.3 Training with Different Number of Constraints

We investigate the empirical training efficiency by using a different number of fairness constraints in our model. We choose BPRMF as our base model to report the training efficiency. Each row in Table 5.3 indicates training with a different number of disparity objectives on the user side and the item side. We observe that our proposed approach has reasonable training time, especially when the number of fairness constraints increases: the more number of constraints added, the less extra time the model needs. This shows the ability of our model to train multiple objectives simultaneously for multiple stakeholders.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We propose a multi-objective optimization framework, *Multi-FR*, for the fairness-aware recommendation in two-sided marketplaces. *Multi-FR* applies the multi-gradient descent algorithm to generate a Pareto set that contains the scaling factor candidate of each objective. Then the Least Misery Strategy is utilized to select the most proper solution from the generated Pareto set by using the Frank-Wolf Solver. To achieve fairness-aware recommendation, four fairness constraints are proposed within the multi-objective optimization framework. Experimental results on three real-world datasets show that our method can constantly outperform various corresponding base architectures and state-of-the-art fair ranking models. Multiple evaluation metrics

clearly validate the performance advantages of *Multi-FR* on the fairness-aware recommendation and demonstrate the effectiveness of the MOO mechanism.

6.2 Future Work

We would like to investigate several points in future work.

First, we plan to investigate the relationship between consumer-sided fairness and producer-sided fairness. In practice, we observe that optimizing the fairness on one side may improve or impair the fairness on the other side. We are interested to see if we can explicitly model this relationship under our definitions. Second, we plan to investigate how to define the platform's utility or fairness in the recommendation, since the platform itself should be another stakeholder in the recommendation. Third, we plan to investigate how to define fairness in a dynamic model, rather than under a static recommendation, since dynamic fairness should be more suitable for a real case.

To the best of our knowledge, these topics are still under-researched in the community. We do think these points are worthy for constructing effective recommendation systems.

Bibliography

Yao Lu, Sandy El Helou, and Denis Gillet. A recommender system for job seeking and recruiting website. In *WWW (Companion Volume)*, pages 963–966. International World Wide Web Conferences Steering Committee / ACM, 2013.

Joseph A. Konstan and John Riedl. Recommender systems: from algorithms to user experience. *User Model. User Adapt. Interact.*, 22(1-2):101–123, 2012.

Chenguang Pan and Wenxin Li. Research paper recommendation with topic analysis. In *2010 International Conference On Computer Design and Applications*, volume 4, pages V4–264–V4–268, 2010. doi: 10.1109/ICCDA.2010.5541170.

Xavier Amatriain, Josep M. Pujol, and Nuria Oliver. I like it... I like it not: Evaluating user ratings noise in recommender systems. In *UMAP*, volume 5535 of *Lecture Notes in Computer Science*, pages 247–258. Springer, 2009a.

Xavier Amatriain, Josep M. Pujol, Nava Tintarev, and Nuria Oliver. Rate it again:

increasing recommendation accuracy by user re-rating. In *RecSys*, pages 173–180. ACM, 2009b.

Sarabjot S. Anand, Patricia M. Kearney, and Mary Shapcott. Generating semantically enriched user profiles for web personalization. *ACM Trans. Internet Techn.*, 7(4):22, 2007.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272. IEEE Computer Society, 2008.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. Recommender systems survey. *Knowl. Based Syst.*, 46:109–132, 2013.

Vipul Vekariya and G. R. Kulkarni. Hybrid recommender systems: Survey and experiments. In *DICTAP*, pages 469–473. IEEE, 2012.

Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In *ICML*, pages 46–54. Morgan Kaufmann, 1998.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009. ISSN 0018-9162. doi: 10.1109/MC.2009.263. URL `http://dx.doi.org/10.1109/MC.2009.263`.

J. Ben Schafer, Dan Frankowski, Jonathan L. Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 291–324. Springer, 2007.

Robin D. Burke. Hybrid recommender systems: Survey and experiments. *User Model. User Adapt. Interact.*, 12(4):331–370, 2002.

Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Syst. Appl.*, 39(11): 10059–10072, 2012.

David Kurokawa, Ariel D. Procaccia, and Junxing Wang. When can the maximin share guarantee be guaranteed? In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 523–529. AAAI Press, 2016.

Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods, 2011.

Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *CIKM*, pages 2243–2251. ACM, 2018.

Nina Baranchuk, Glenn MacDonald, and Jun Yang. The economics of super managers. *The Review of Financial Studies*, 24(10):3321–3368, 2011.

Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search.

Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Jul 2019. doi: 10.1145/3292500.3330691. URL http://dx.doi.org/10.1145/3292500.3330691.

Gourab K. Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *WWW*, pages 1194–1204. ACM / IW3C2, 2020.

Mifa Kim, Tomoyuki Hiroyasu, Mitsunori Miki, and Shinya Watanabe. SPEA2+: improving the performance of the strength pareto evolutionary algorithm 2. In *PPSN*, volume 3242 of *Lecture Notes in Computer Science*, pages 742–751. Springer, 2004.

A. Ghane-Kanafi and E. Khorram. A new scalarization method for finding the efficient frontier in non-convex multi-objective problems. *Applied Mathematical Modelling*, 39(23):7483–7498, 2015. ISSN 0307-904X. doi: https://doi.org/10.1016/j.apm.2015.03.022. URL https://www.sciencedirect.com/science/article/pii/S0307904X15001742.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*, pages 525–536, 2018.

Rishabh Mehrotra, Amit Sharma, Ashton Anderson, Fernando Diaz, Hanna Wallach, and Emine Yilmaz. Auditing search engines for differential satisfaction across demographics. In *Proceedings of the 26th International Conference on World Wide Web*, 2017.

Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 172–186, New York, NY, USA, 23–24 Feb 2018a. PMLR.

Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 224–232, New York, NY, USA, 2018. Association for Computing Machinery.

Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 2925–2934, Red Hook, NY, USA, 2017. Curran Associates Inc.

Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *The 41st International ACM SIGIR*

Conference on Research & Development in Information Retrieval, SIGIR '18, pages 405–414, New York, NY, USA, 2018. ACM.

Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5427–5437. Curran Associates, Inc., 2019.

Fernando Diaz, Bhaskar Mitra, Michael D. Ekstrand, Asia J. Biega, and Ben Carterette. Evaluating stochastic rankings with expected exposure. In *CIKM*. ACM, 2020.

Michael D. Ekstrand, Mucun Tian, Mohammed R. Imran Kazi, Hoda Mehrpouyan, and Daniel Kluver. Exploring author gender in book rating and recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 242–250, New York, NY, USA, 2018b. Association for Computing Machinery.

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. Fairness in recommendation ranking through pairwise comparisons. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis, editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2212–2220. ACM, 2019. doi: 10.1145/3292500.3330745. URL https://doi.org/10.1145/3292500.3330745.

Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. Balanced neighborhoods for multi-sided fairness in recommendation. In *Conference on Fairness, Accountability and Transparency*, pages 202–214, 2018.

Tom Sühr, Asia J Biega, Meike Zehlike, Krishna P Gummadi, and Abhijnan Chakraborty. Two-sided fairness for repeated matchings in two-sided markets: A case study of a ride-hailing platform. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3082–3092, 2019.

Tamas Jambor and Jun Wang. Optimizing multiple objectives in collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, page 55–62, New York, NY, USA, 2010a. Association for Computing Machinery. ISBN 9781605589060. doi: 10.1145/1864708.1864723. URL `https://doi.org/10.1145/1864708.1864723`.

Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI Extended Abstracts*, pages 1097–1101. ACM, 2006.

Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. Fairness-aware group recommendation with pareto-efficiency. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, page 107–115, New York,

NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346528. doi: 10.1145/3109859.3109887. URL https://doi.org/10.1145/3109859.3109887.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *KDD*, pages 2219–2228. ACM, 2018.

Marco Tulio Ribeiro, Anisio Lacerda, Adriano Veloso, and Nivio Ziviani. Pareto-efficient hybridization for multi-objective recommender systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, page 19–26, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312707. doi: 10.1145/2365952.2365962. URL https://doi.org/10.1145/2365952.2365962.

Marco Tulio Ribeiro, Nivio Ziviani, Edleno Silva De Moura, Itamar Hata, Anisio Lacerda, and Adriano Veloso. Multiobjective pareto-efficient approaches for recommender systems. *ACM Trans. Intell. Syst. Technol.*, 5(4), December 2015. ISSN 2157-6904. doi: 10.1145/2629350. URL https://doi.org/10.1145/2629350.

Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. Privacy-aware recommendation with private-attribute protection using adversarial learning. *Proceedings of the 13th International Conference on Web Search and Data Mining*, Jan 2020. doi: 10.1145/3336191.3371832. URL http://dx.doi.org/10.1145/3336191.3371832.

Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards

fair graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3289–3295. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/456. URL `https://doi.org/10.24963/ijcai.2019/456`.

Avishek Joey Bose and William L. Hamilton. Compositional fairness constraints for graph embeddings, 2019.

Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. Fa*ir: A fair top-k ranking algorithm. In *CIKM*, pages 1569–1578. ACM, 2017.

L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with fairness constraints. In *ICALP*, volume 107 of *LIPIcs*, pages 28:1–28:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

G. Birkhoff. *Lattice Theory*. American Mathematical Society, Providence, 3rd edition, 1967.

K. Deb, K. Sindhya, and J. Hakanen. Multi-objective optimization. In *In Decision Sciences: Theory and Practice*, pages 145–184. CRC Press, 2016.

Tamas Jambor and Jun Wang. Optimizing multiple objectives in collaborative filtering. In *RecSys*, pages 55–62. ACM, 2010b.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second*

Berkeley Symposium on Mathematical Statistics and Probability, pages 481–492, Berkeley, Calif., 1951. University of California Press. URL `https://projecteuclid.org/euclid.bsmsp/1200500249`.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 427–435. JMLR.org, 2013.

Toon De Pessemier, Simon Dooms, and Luc Martens. Comparison of group recommendation algorithms. *Multim. Tools Appl.*, 72(3):2497–2541, 2014.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback, 2012.

Ziwei Zhu, Xia Hu, and James Caverlee. Fairness-aware tensor-based recommendation. In *CIKM*, pages 1153–1162. ACM, 2018.

Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *CHI*, pages 3819–3828. ACM, 2015.

Amelia Butterly. Google image search for ceo has barbie as first female

result. 2015. URL http://www.bbc.co.uk/newsbeat/article/32332603/google-image-search-for-ceo-has-barbie-as-first-female-result.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, 2016.

R. Plackett. The analysis of permutations. 1975.

Alistair Moffat and Justin Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Trans. Inf. Syst.*, 27(1), December 2008. ISSN 1046-8188. doi: 10.1145/1416950.1416952. URL https://doi.org/10.1145/1416950.1416952.

Mingrui Wu, Yi Chang, Zhaohui Zheng, and Hongyuan Zha. Smoothing DCG for learning to rank: a novel approach using smoothed hinge functions. In *CIKM*, pages 1923–1926. ACM, 2009.

Tao Qin, Tie-Yan Liu, and Hang Li. A general approximation framework for direct optimization of information retrieval measures. *Inf. Retr.*, 13(4):375–397, 2010.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables, 2016.

Sebastian Bruch, Shuguang Han, Michael Bendersky, and Marc Najork. A stochastic treatment of learning to rank scoring functions. In *WSDM*, pages 61–69. ACM, 2020.

Gini Index, pages 231–233. Springer New York, New York, NY, 2008. ISBN 978-0-387-32833-1. doi: 10.1007/978-0-387-32833-1_169. URL https://doi.org/10.1007/978-0-387-32833-1_169.

Measurement of Diversity, pages 688–688. Nature, 1949. doi: 10.1038/163688a0. URL https://doi.org/10.1038/163688a0.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461. AUAI Press, 2009.

Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, pages 165–174. ACM, 2019.

Sylvain Bouveret, Yann Chevaleyre, Nicolas Maudet, and Hervé Moulin. *Fair Allocation of Indivisible Goods*, page 284–310. Cambridge University Press, 2016. doi: 10.1017/CBO9781107446984.013.

Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *IJCAI*, pages 91–97. ijcai.org, 2018.

Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah,

and Junxing Wang. The unreasonable fairness of maximum nash welfare. *ACM Trans. Economics and Comput.*, 7(3):12:1–12:32, 2019.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.