Algorithmc bias in Recommender Systems

Investigating the behavior of Recommender Systems bias and fairness interventions

Petar Petrov

TU Delft

Algorithmc bias in Recommender Systems

Investigating the behavior of Recommender Systems bias and fairness interventions

by

Petar Petrov

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday June 17, 2026 at 13:30.

Student number:	5215781	
Project duration:	September 1, 2026 – June 17, 2026	
Thesis committee:	Prof. dr. Alan Hanjalic ,	TU Delft, Chair
	Dr.Masoud Mansoury,	TU Delft, Advisor
	Dr. Avishek Anand,	TU Delft, Core Member 2

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Summary

Recommender systems have seen considerable adoption in recent years, driven by modern streaming services, social media, and novel LLM applications. However, recommender systems show clear statistical biases and can expose stakeholders to social biases. This phenomenon is caused by several factors, such as the data, which tends to be too dirty for the statistical methods used in recommendation; the pipeline, which might directly introduce bias into recommendations; and the evaluation, which is often unaware of the underlying biases in the task. As such, we first reviewed current literature on the topic and discovered several discrepancies. Firstly, fairness-aware solutions, which aim to reduce social bias, are often not tested on Missing-at-Random (MAR) data, which is cleaner than the traditional Missing-not-at-Random (MNAR) data used in recommendations. Further, we establish a connection between statistical bias and social bias, and identify the need for user-group-based studies. As such, we first tested whether fairness-aware solutions benefit from MAR data similarly to debiasing solutions, which aim to reduce statistical bias. Then we investigated the extent to which debiasing solutions can address fairness issues, before finally delving into more detail on the individual user-group performance of some of our configurations. We found that some fairness-aware algorithms benefit from MAR data, though this does not appear to be universal. We also observed a noticeable benefit from diversification enabled by debiasing solutions, and we identified interesting insights into how interventions impact users based on the share of popular items they interact with.

Contents

List of Figures

List of Tables

1

Introduction

Machine Learning (ML) has seen wide adoption in recent years due to its ability to support time series processing, image classification, and other complex tasks that traditional algorithms struggle with. However, this adoption has exposed a large issue at the core of ML. Machine Learning algorithms suffer from inherent bias, which can degrade their accuracy and harm stakeholders in complex systems. For example, recent facial recognition systems have shown reduced accuracy for non-white individuals [9], meaning that large portions of the population are likely to experience negative interactions with such systems. This phenomenon is not exclusive to image classification either. Research first showed that machine translation models exhibit gender bias a decade ago, with more recent work suggesting that solutions still fail in some areas [32].

Widescale adoption has also exposed fundamental ethical concerns in the formulation of some ML tasks. At their core, Machine Learning algorithms create an approximation of the training data that is flexible enough to account for new data. However, this makes algorithms highly sensitive to the data they ingest. For example, if we were to train a crime monitoring system on a dataset primarily consisting of incidents in a specific part of town, or involving a specific ethnic or religious group, the model would learn an association between these data features and its tasks. Simply put, the model could learn to directly associate an irrelevant feature of users/items/data with the task. We call data with such clear biases biased data. Not only will this hurt the model's performance, but in a serious enough use case, it can lead to disastrous consequences for users. Automated crime monitoring systems trained on biased data can reinforce negative racial stereotypes, while an automated investment system might hyperfocus on poor but shortsighted investments due to their prevalence in a biased dataset. Biased data can even expose system manufacturers to lawsuits if a system violates a user's rights.

Recommender Systems (RS, RecSys), like many other fields nowadays, have adopted Machine Learning with open arms. One of the most common recommender formulations, Collaborative Filtering (CF), leverages modern Machine Learning and Deep Learning (DL) techniques to learn embeddings such that the dot product between them reveals user preferences. As such, the problem of fairness and bias in ML extends to Recommender Systems too. Furthermore, research has consistently shown that recommenders exhibit various biases, as discussed in Section 2. At the same time, we can see recommenders employed in more and more fields, thanks to advances brought on by Large Language models (LLMs). For example, many LLM chatbots use a Retrieval-Augmented-Generation (RAG) pipeline to ensure factual responses. Bias and fairness have also been topics in RS for a long time, with various forms of bias shown in research. For example, as we will discuss, Zehlike et al. [48] presents a scenario in which job candidate recommendations can lead to biased outcomes for women. Other papers have identified several groups of bias and fairness issues that can arise in a recommendation pipeline [7, 43].

Recommender systems have gained a considerable amount of interest in recent history due to the proliferation of various modern platforms such as social media and streaming services. Recommenders enable us to sift through large amounts of data efficiently and easily, identifying individual items relevant

to users. Such systems get deployed to create curated feeds on social media platforms, allowing users to interact with like-minded individuals and discover new, relevant posts. Similarly, streaming services have had to adopt recommendation algorithms to help users find engaging content on their platforms.

In the traditional recommendation pipeline, an algorithm is designed and evaluated offline against a relevant dataset consisting of user-item interactions. More formally, the dataset contains an interaction record for each user $u \in U$ and item $i \in I$ if u has interacted with i. Interaction data is commonly represented as a matrix with specific interaction information for each user-item pair, such as a binary interaction flag (where 1 indicates an interaction and 0 indicates its absence) or a rating on a standard scale. As in other ML setups, the dataset is often split into three subsets: training, test, and validation. The task is then to learn a model function f' that maximizes accuracy on the unseen test set, while training on the typically much larger training set, with validation serving as a tool for hyperparameter optimization or a secondary performance evaluation.

Although this setup is quite successful, recent research has exposed issues. Namely, recommenders exhibit algorithmic bias and unfair predictions, which negatively impact the algorithm's accuracy and introduce undesirable outcomes in the system's outputs. While bias is often an unintended byproduct of a pipeline's design, fairness is an ethical decision made by the designers; as such, the relationship between fairness and bias in recommender systems remains unexplored. Properly distinguishing between the two can often be quite challenging because both phenomena result in the same unexpected behavior.

One of the most common types of bias observed in Recommender systems is popularity bias, where users are recommended popular items based solely on their existing popularity, rather than their relevance to the user [37]. Besides reduced accuracy, popularity bias can make the algorithm's outcomes unfair by disproportionately recommending items with existing popularity, effectively suppressing items that are yet to be 'discovered'. In critical scenarios, such as medical treatment recommendations, popularity bias can have disastrous effects on the system's outcome by prioritizing popular treatments over those that are more relevant for the patient.

We can better understand the effects of bias on recommendations by evaluating a pipeline in an unbiased manner, typically using specialized metrics or debiased data. Debiased data is created by randomly prompting users with items instead of relying solely on past interactions [6]. Reliance on past interactions from existing recommendation systems risks introducing undefined behavior in a novel pipeline, as it mirrors the biases of the pre-existing system. One of the most significant issues with past interactions is that ratings missing from the data are missing-not-at-random (MNAR), meaning we must account for external factors when using the data. In contrast, interactions in a debiased dataset are selected at random, making missing ratings missing-at-random (MAR) and removing the need for us to consider external factors. As such, debiased data is less noisy and better suited to the task at hand [6]. Because of this, approaches for reducing bias are sometimes evaluated exclusively against debiased test sets, thereby minimizing the effect of pre-existing, undefined behavior in the data. However, we rarely do this when improving fairness, which may misrepresent the effectiveness of fairness approaches, as we estimate performance on data that may not be entirely fair.

Metrics have been a consistent issue on the evaluation front as well. Many recommendation metrics initially originated in Information Retrieval (IR). Although there is overlap, these metrics fail to accurately represent the recommender's performance due to differences between the IR and RS tasks [41]. At the same time, metrics that target recommenders directly, while more accurate, do not always account for known biases in the task. Another reason we rarely see metrics that account for bias and fairness is the overreliance on accuracy in modern RS research, which creates pipelines that appear more accurate but may exhibit exaggerated, undefined behavior due to a lack of research on the behavior of the proposed solution.

Finally, unexpected behavior can also occur in the algorithm's design directly by taking advantage of the biased properties of the data or the overall task. Such models achieve good performance, at the cost of highly skewed results that ultimately fail to represent the underlying task. While their performance might be impressive, they might not perform as well long-term and might fail to account for long-tail relationships that are important for Recommendation. Another cause of undefined behavior could be unnecessary complexity in the optimization task, often due to attempts to correct possible bias and

fairness issues. In such cases, we frequently employ multi-objective optimization schemes, which have demonstrated greater success in optimizing for multiple distinct goals. However, MOOs struggle with gradient issues and can fail to determine the correct objective to optimize against, resulting in difficulties in finding an optimal solution to the recommendation task.

Due to the numerous moving parts that comprise a recommendation pipeline, it can often be challenging to pinpoint the cause of undefined behavior in the outputs. At the same time, research focuses on increasing accuracy while treating bias and fairness as secondary factors; this leads to highly accurate recommendation pipelines, but often on data that is itself not truly representative of the goal of the task, using metrics that fail to evaluate recommendations properly, and/or by exploiting facts about the data that hurt the performance on clean, unseen interactions. Collecting more data about the behavior of solutions at various points in the recommendation pipeline could help us better understand the interplay between bias, fairness, and accuracy. It will allow us to determine how to optimize for all of them.

1.1. Research questions and contributions

In this paper, we aim to gain a better understanding of the interplay between bias and fairness in recommender systems by systematically exploring the prevalence of several types of bias and fairness issues through a comprehensive evaluation scheme. We further aim to answer the following research questions.

Firstly, **"What trends in accuracy can we observe when applying fairness-aware solutions on MAR data?"**. During our exploratory phase, we identified that by and large, fairness-aware algorithms are rarely evaluated against MAR data. At the same time, research has shown that MAR data provides a more accurate and realistic estimate of an algorithm's performance. As such, we aim to investigate the performance of various fairness-aware algorithms on MAR data.

Secondly, **"What trends can we observe in the effects of modern RS debiasing optimizations on fairness?"**. We noticed that, in research, debiasing algorithms are typically evaluated only for their accuracy and/or debiasing impact. However, we also believe it is valuable to investigate their fairness outcomes, as there appears to be a connection between bias and fairness in Recommender systems, given that many debiasing and fairness-aware algorithms use overlapping definitions of the negative impact they aim to minimize. As such, we hope that by investigating the fairness outcomes of debiasing algorithms, we can better understand how well they perform. Furthermore, we hope to learn exactly how connected debiasing and fairness-correction are.

Finally, research typically reports only aggregated dataset-level metrics of an algorithm's performance. However, an intervention or novel recommender pipeline might impact different users differently. As such, we aim to answer the research question **"What is the impact of bias and fairness intervention on different user groups?"** by segregating users by the share of popular items they have interacted with, allowing us to examine the impact of interventions on users based on their preferences.

Contributions

This work makes the following contributions: first, we include additional analysis across several vectors, in the hope that future system designers can make more informed decisions when constructing novel pipelines. Specifically, alongside additional **information on each intervention's impact on bias and fairness**, we provide a **breakdown of user group performance**. Furthermore, we contribute additions to the RecBole framework, which we use for our experiments. In particular, we contribute **extendable versions of the intervention approaches** discussed in this work, alongside supporting code. We hope that these additional implementations can aid future research in better understanding the behavior of recommender systems interventions.

1.2. Structure

The rest of this theis is structured as follows: in chapter 2 we provide necessary background information about our work, discussing bias and fairness in Recommender systems, and approaches for mitigating the effects of bias and unfairness. Then in chapter 3 we present our problem statement alongside our experimental setup discussing the algorithms, details, and particularies about the evaluation setup. We present results in chapter 4 before finally discussing them in chapter 5. We close everything off with a

final conclusion based on our research chapter 6.

2

Related work

In this chapter, we review related work and lay the groundwork for our experiments. First, we discuss bias in recommender systems before focusing on fairness. Then we break down several intervention approaches to address bias and fairness in pipelines.

2.1. What even are bias and fairness in RecSys?

In this section, we discuss bias and fairness. First, we introduce the concepts, discuss their relevance to Machine Learning and Computer Science broadly, and then delve deeper into how they impact recommender systems.

2.1.1. Bias

Intuitively, when people hear bias, they think of systemic issues such as the gender pay gap and unequal treatment by the judicial system of individuals from underrepresented social groups. In the case of recommendation, this can manifest as the systematic misrepresentation of underrepresented items. For example, a hiring platform exposed to data that primarily shows male candidates as suitable might overexpose male candidates relative to female ones [17]. While there are works that discuss this facet of bias in recommender systems, they often present it as a fairness issue rather than bias. RecSys, like many other ML-dominated fields, relies on statistical algorithms and tools that approximate a function from existing data. These approaches entail certain expectations that, if unmet, introduce statistical biases into the model's output. One example of this is the data, which is typically MNAR (missing not at random), thereby introducing statistical bias into the pipeline [33], as the data is not truly representative of the underlying population.

While the two definitions are conceptually different, they both fall under the label of algorithmic bias, as they both manifest as effects of the recommendation pipeline. However, the way we correct each of these phenomena can differ greatly. Statistical bias can be factored directly into the pipeline's formulation through specialized model structures, loss functions, and training schemes, allowing the recommender to actively debias its recommendations during training and inference. Still, social bias can be difficult to quantify because it lacks a purely statistical definition, unlike biases such as popularity bias, selection bias, etc. Social bias inherently depends on the task at hand, the data, and the system's application, making it much more difficult to identify who might be affected. The same pipeline that recommends movies on Netflix can also learn to recommend medical treatments, yet both applications involve completely different ethical considerations.

Furthermore, confusion over the use of terms, as documented by Lipton and Steinhardt [18] in the ML field, creates scenarios in which it is difficult to determine an algorithm's goal. For example, an intervention might aim to correct a specific bias by applying a social view, such as equal distribution of a specific quality (e.g., item exposure, custom metrics, etc.) across a group of items or users, thereby transforming the intervention's goal from correcting statistical bias to addressing a social effect. At the same time, statistical bias can have a negative social impact depending on how the recommenda-

tion system is applied. For example, popularity bias can cause music recommendation platforms to underrepresent minority artists, genres, and gender groups [16].

As previously discussed, one of the most common biases in recommender Systems is popularity bias. Examining the rating-item distribution reveals that we can view popularity bias as a type of selection bias, in which a skewed item probability distribution influences selection. In this case, popularity bias occurs when the likelihood of ratings depends solely on the item to which they correspond. This formulation enables us to identify positivity bias, in which the probability of a rating depends solely on its value, with higher ratings corresponding to higher propensities. Where popularity bias exaggerates the popularity of items out of proportion [21], positivity bias reinforces the dominance of popular items [26]. However, these are not the only types of bias. Research such as Chen et al. [7] further identifies four more types of bias. Specifically, they identify Exposure bias, Conformity bias, Position bias, and Inductive bias. Exposure bias occurs when unobserved interactions are treated as negative, even though a user might simply not have been exposed to the item due to the pipeline's existing biases. Conformity is largely driven by user behavior, as users align with existing opinions about an item's popularity. For example, some users might watch a movie on a streaming platform simply because others have seen it, rather than because they find it compelling or interesting. Position bias is caused by users' tendency to prefer the first few items, and Inductive bias, as identified by the authors, is explicitly introduced by researchers or engineers to improve recommendations by generalizing better or reducing recommendation time. Finally, the authors identify unfairness as an explicit form of bias and correctly point to a connection between popularity bias and fairness.

Huang et al. [14] also introduce a multifactorial framework for estimating selection bias, making a compelling case for the multifactorial nature of bias in recommender systems and effectively motivating their solution. However, the authors observed that their multifactorial solution can struggle to learn correctly, necessitating more complex optimization strategies. Notions of bias can be embedded directly into metrics by exploiting item popularity information, such as the number of occurrences of a given item in the training data, allowing us to easily integrate estimates of a pipeline's performance with respect to a specific bias.

2.1.2. Fairness

Fairness has been a topic of interest to people since Ancient Greece [2]. In recent history, modern democracies have embedded notions of fairness into their legal and governmental systems to ensure equitable treatment and representation. While we do see fairness applied in scientific fields to ensure higher-quality data and more generalizable studies, there are still numerous ethical issues in modern research. Biases in experiments limit the extent to which results generalize, making it harder for researchers to draw proper conclusions and hurting the overall value of the work.

For example, studies have shown that Psychology research draws largely on samples from industrialized, wealthy Western nations [11]. However, while research tends to assume that these samples generalize across the whole human population, analysis has shown that the behavior of people from such nations is not truly representative of the global population [3]. We see a similar situation in medical research, where the majority of participants in clinical trials and other studies are male [42]. Meaning that, research tends to misrepresent women. While at first thought, this might not be a problem, and medical research tends to treat men and women similarly, research has shown that women can have different reactions to drugs and treatments [20]. The lack of fair representation ultimately creates an environment in which treatments may be applied blindly, without a full understanding of their potential side effects. A striking example of this is that, reportedly, women are 50 to 75% more likely to experience adverse drug reactions [27].

We can use modern technology to alleviate some of these issues. Our world is ever more connected, allowing us to collect samples from various parts of the world, providing a more diverse and fair representation of the population for studies. At the same time, researchers can use algorithmic tools to understand better how drugs behave before trials even begin, allowing them to determine the types of participants needed for their studies. However, the adoption of modern technology has not been without its own ethical issues. For example, several governments restrict the kinds of information their citizens can consume, instead presenting an approved view of the world. Novel surveillance technology and Machine Learning techniques have made comprehensive data protection laws a necessity.

Research has already identified various forms of unfairness [32] in ML. However, most works focus on improving accuracy and presenting novel solutions rather than addressing issues with existing systems. Often, systems are released without prior consideration of potential ethical violations, leading to negative experiences for users [17, 9, 32].

As one of the fields that has seen wide adoption in recent history, Recommender Systems can be at the center of some of these ethical concerns. For example, research has shown that job recommendation platforms are biased against female candidates by preferring male candidates [17]. Similarly, modern streaming platforms use recommenders to sift through the massive libraries of content they own and only expose relevant items to users. However, these recommendations can suffer from popularity bias [40], where the system recommends items to users based on popularity rather than relevance. Recommenders can also make it more difficult for smaller-scale sellers to compete on online marketplaces by prioritizing products from well-known, established sellers [35]. While some of this behavior is not surprising, as we would expect many users to conform to popular interest and to prefer established vendors to novel ones, recommenders have the unique ability to reinforce this behavior by exposing biases in their data, which primarily consists of past interactions. Simply put, if the recommender's past recommendations were unfair to begin with, the interactions we record will not be fair either.

Fairness is harder to detect because designers often introduce fairness objectives to account for ethical or external considerations. Simply put, it is difficult to formulate a metric that accounts for all kinds of fairness. For example, a hiring candidate recommendation platform should present each candidate fairly based on their abilities rather than on factors such as gender. At the same time, an online shopping platform would benefit from equalizing exposure for all users, as this guarantees greater exposure for item providers and higher satisfaction.

As a result, unfairness detection algorithms typically adopt a specific definition of fairness or bias and build an algorithm around it. For example, Biega, Gummadi, and Weikum [4] introduced amortized fairness based on position bias. Position bias disproportionately harms low-ranked items, as users tend to interact more with the top items in a list. The authors estimate the attention a series of items receives and attempt to make it proportional to the accumulated relevance. Similarly, Zhu et al. [55] introduces fairness objectives centered on popularity-opportunity bias and presents distinct metrics for item and user popularity-opportunity bias, alongside two approaches to correcting the bias. The authors define popularity-opportunity bias as the phenomenon in which, given two relevant items, the more popular item is ranked higher in the recommendation list than the less popular one. Given this definition, the authors formulated two metrics, PRU and PRI, based on Spearman's rank correlation. PRU measures what the authors describe as uPO bias, which concerns whether rankings are correlated with popularity, a phenomenon quite similar to position bias. PRI, which aims to measure iPO (item popularity-opportunity) bias, assesses whether an item's recommendation position is correlated with its popularity. We show the equations for these metrics below:

$$PRU = -\frac{1}{N}\sum_{u\in\mathbb{U}} SRC(pop(\bar{O_u^+}), rank_u(\bar{O_u^+})) \tag{2.1}$$

$$PRI = -SRC(pop(I), avg\_rank(I)) \tag{2.2}$$

There have also been studies aimed at understanding the different definitions of fairness discussed in recommender systems research. Specifically, Wang et al. [43] establishes a detailed taxonomy of fairness definitions. First, they identify two broad groups based on the recommendation process and the outcome. Specifically, Process fairness aims to ensure a fair recommendation process, while Outcome fairness concerns the fairness of the recommendation scores. Because the majority of recommender systems research on this topic focuses on the fairness of the pipeline's results, the authors then divide Outcome fairness into six distinct groups. First, they identify notions of fairness that focus on ensuring equitable outcomes across a specific group: Individual fairness, where we expect similar individuals to be treated similarly, and Group fairness, where we expect the same treatment across a specific fairness-aware attribute, such as gender, age, or race. The authors then identify a second group centered on unique concepts of fairness. For example, this includes Calibrated fairness, which encompasses a wide range of fairness interventions, including the previously mentioned approaches introduced by

Biega, Gummadi, and Weikum [4] and Zhu et al. [55]. Calibrated fairness requires that the measured value of recommendations be proportional to their merit. Finally, the authors define some less common notions of fairness. This includes Counterfactual fairness, which means that if an individual's group or outside quality changed, the outcome will not change, and Maximin-shared Fairness, which requires all groups or individuals to receive better outcomes than their maximum share. The authors also identify Consistent Fairness as a unique form within the Concept Fairness group, but from their explanations and justifications, it appears to be quite close to Individual Fairness in both function and definition.

We can also approximate fairness outcome using traditional statistical notions of fairness. For example, the Gini Index, which is already widely used as a fairness metric for evaluating income or wealth inequality, can also estimate recommendation unfairness by indicating how similar the distribution of recommended items is to the distribution in the training data. Another example is Item Coverage, which directly indicates how much of the item catalog we expose users to. While not originally designed for recommendation, such metrics serve as valuable proxies that can help us understand a pipeline's behavior. Furthermore, because they do not embed a pre-existing notion of fairness into their definition, they serve as more neutral estimators than purpose-built metrics.

2.2. Mitigation approaches

In this subsection, we discuss approaches to mitigate undefined behavior in the Recommender Systems pipeline.

2.2.1. Pre-processing

We can separate dataset-level mitigation approaches into two broad categories: debiased datasets and intervened datasets. Debiased datasets are created to provide unbiased data by randomly exposing users to items rather than relying on a recommender to drive feedback, whereas intervened datasets attempt to correct for bias when forming batches.

Missing interactions are often missing-not-at-random (MNAR), introducing many of the biases we later have to correct, because they are missing due to a systematic issue, such as the collection process, failures in a pre-existing recommender, or a quirk of the system. In contrast, debiased datasets prompt users with random items, thereby building a dense collection of interaction data where values are missing at random (MAR). MAR data is more conducive to machine learning than MNAR data, as it does not require accounting for external factors involved in data collection. Popular examples of debiased datasets include Yahoo R3 [19], and Coat [33]. Unfortunately, debiased datasets are rare due to the time and financial commitment necessary to create them. As a result, research also examines sampling approaches that aim to mitigate bias in existing data.

Sampling approaches use statistical tools and analysis to construct representative samples from biased data. One such approach is SKEW [47], which samples user-item interactions in inverse proportion to item popularity, directly reducing popularity bias. However, the authors do not directly test the empirical value of their work. Carraro and Bridge [6] introduces WTD, an intervention strategy that utilizes weights to make the intervened dataset resemble MAR data in terms of user-item posterior probabilities. The authors note that WTD requires MAR data to accurately calculate the weights and introduce an alternative formulation, WTD_H, that assumes the posterior probability distribution for MAR data is uniform and uses this assumption to calculate the weights. The authors provide empirical evidence that intervened datasets can be used for debiasing, demonstrating that SKEW, WTD, and WTD_H yield results similar to those from purely MAR data. We should note that WTD and WTD_H were devised as tools for debiasing test sets rather than training sets. However, the authors argue that their solution is suitable for generating both training and testing data by relying on the hypothesized posterior probability distribution.

The above-discussed solutions focus primarily on statistical bias. However, we can also identify preprocessing interventions to address fairness issues in research. For example, Ekstrand et al. [8] conducted a study on the effectiveness of recommendations across different user demographic groups. The authors identify a clear imbalance in the gender distribution across both datasets they consider and, to correct it, randomly resample the datasets so that each gender has an equal number of samples. The authors note that, overall, random resampling improved the pipeline's outcome across groups only

marginally, while also reducing accuracy. Inspired by data-poisoning attacks Rastegarpanah, Gummadi, and Crovella [28] introduced a preprocessing unfairness intervention approach that generates ”antidote” data based on a socially-aware objective function. In this context, ”antidote” data is data that affects the predicted rating matrix, thereby modifying the fairness outcome of the recommendations. This approach, while showing good results, requires training to optimize the antidote data using gradient descent, making it very time-consuming.

2.2.2. In-processing

Fairness and bias-aware algorithm design has already made its way to the RS field. We often use Multi-Objective Optimization (MOO) approaches, which enable models to achieve high accuracy while meeting specific bias and fairness constraints. Paparella et al. [22] presents a summary of the state-of-the-art MOOs approaches for Recommendation Systems, alongside information about the reproducibility of the covered works. In particular, the authors focus on Stamenkovic et al. [36], which introduced SMORL, a Scalarized Multi-Objective Reinforcement Learning framework aiming to satisfy accuracy, diversity, and novelty in session-based recommenders. Although SMORL delivers in terms of accuracy, diversity, and novelty, Paparella et al. [22] note that SMORL might have unintended effects on downstream bias, possibly worsening performance in categories not covered by the task.

We can also find justification for treating recommendation as a combination of multiple tasks in the formulation of existing, popular recommenders. For example, Hsieh et al. [13] recast collaborative filtering as a metric learning objective, integrating a metric learning task alongside the existing recommendation task. While the authors do not directly identify this as an MOO solution, they employ a standard metric learning loss combined with rank-aware weighting, effectively introducing two tasks into the model’s objective and opening the door for an MOO approach. This paper is a good example of how the overreliance on accuracy in new pipeline formulations can be problematic. The authors only report Recall@50 and Recall@10 to justify their design, failing to recognize how popularity bias can influence their outputs through their assumptions. For example, the researchers argue that the triangle inequality benefits recommendation as: 1) it clusters users who co-like the same item together, and 2) the items co-liked by the same users are clustered together. However, this can also cause popularity bias by boosting popular items, which are often liked by multiple users, even further.

Another common approach is to leverage causal techniques to achieve fairness and bias constraints [50, 54, 44]. Specifically, causal approaches use structural causal models and do-calculus to identify the true causal impact of recommending an item. Such approaches typically focus on constructing causal embeddings that aim to remove popularity bias, providing a recommender with cleaner, simpler embeddings to use during recommendation. For example, Zhang et al. [50] establishes a causal link between popularity and recommendation using do-calculus. They argue that popularity affects an item’s exposure as recommenders show inherent bias towards popular items. At the same time, the authors argue that we cannot fully remove popularity bias due to herd mentality, meaning that we will hurt the recommender’s performance if we debias too far, as users exhibit conformity that leads them to follow the majority in consuming popular items. Based on these observations, the authors introduce Popularity Deconfounding and Adjustment (`PDA`), which attempts to remove bias from the training task by optimizing the widely used Bayesian Personalized Ranking (BPR) loss, then reintroduces bias during evaluation by scaling the model’s scores by an item’s popularity.

There appears to be multiple angles through which we can apply causal inference in recommender systems. For example, Wei et al. [44] also use causal inference to derive a debiased framework, but arrive at a different formulation. Specifically, they introduce independent item and user modules that aim to ”soak up” bias, which can then be removed during inference. Simply put, the modules are trained to estimate popularity bias within their respective groupings (user vs. item), providing an estimate of the expected bias. The proposed Model-Agnostic Counterfactual Reasoning (`MACR`) model then removes the estimated bias from the scores during inference, which the authors support through a do-calculus derivation.

A common, and nearly ubiquitous, way to correct bias is through regularization. Inverse Propensity Scoring (`IPS`) [33] is possibly the most common of these approaches. `IPS` weights each item’s score inversely to its propensity. The propensity score is the probability of observing a given item-user pair [15], so item-user pairs with a high likelihood are less common. In contrast, those with a low probabil-

ity become more common. Although IPS has shown great success in accounting for popularity bias, among other biases too [14], the method of estimating propensities remains a bottleneck, as we require high-quality data to build a realistic propensity distribution. The broader idea of using the inverse probability of a rating has also spread, as seen in Xu et al. [47] and Carraro and Bridge [6]. Propensities make up the backbone of many in-processing interventions, including Zhang et al. [50]. However, they are not the only way to regularize a pipeline's recommendations. Zhu et al. [55] introduces an alternative approach centering around Pearson's Correlation Coefficient. Essentially, the authors regularize the scores based on the correlation coefficient between the predicted scores for user-item pairs and the corresponding item popularity.

2.2.3. Post-processing

Another common approach is to directly modify a pipeline's recommendations after scoring [48, 5, 55]. For example, Zehlike et al. [48] presents a job-hiring example, focusing on the desperate outcomes users might experience due to an imbalanced gender distribution. The authors' solution, FA*IR, explicitly includes protected items in the recommendation list based on a minimum ratio and a significance test. In particular, the algorithm creates a list of protected and non-protected items from a set of recommendations, with classification based on item quality (e.g., gender, ethnic background, product category, etc.). The algorithm then reconstructs the recommendation list, while ensuring that a specific number of protected items are available at each rank.

We can improve the outcome of recommendations without establishing a protected list, too. For example, Biega, Gummadi, and Weikum [5] introduces an approach in which we calculate an objective measure of the engagement an item or user receives and adjust the recommendation scores to equalize it. Zhu et al. [55] builds on this idea further by applying it to popularity-opportunity bias. Specifically, the authors present a solution that calculates the required compensation for each item using three guidelines. The guidelines aim to promote low-popularity items while also maintaining user preferences and ensuring we do not blindly apply compensation to all low-popularity items. Finally, the third guideline aims to provide equal intervention for all users by compensating users with large value scales accordingly.

Interestingly, most post-processing solutions focus on fairness and social bias. As such, Wang et al. [43] split reranking solutions into three categories based on the kind of reranking they do. Slot-wise reranking selects items for the final recommendation list, one at a time. Serbos et al. [34] and Zehlike et al. [48] are both slot-wise algorithms that use a greedy approach to reconstruct a fair recommendation list for users. User-wise reranking aims to find the best ranking list for a user based on an objective. Unlike slot-wise rerankers, which tend to use greedy algorithms, many user-wise solutions use integer programming [45, 5] to abstract the decisions the rerankers make and to impose constraints on the reranking problem that align with the design goals. In contrast, globally-wise reranking aims to rerank across multiple users and recommendation lists simultaneously. While some global-wise reranking solutions also use mathematical programming concepts [39], there is a considerable amount of intervention formulations that involve more complex, two-step processes [23] or distribute items based on a calculated utility/effort/compensation/etc [55].

3

Methodology

This chapter outlines our problem statement and experimental design. First, we discuss the Problem statement to establish the problems we are concerned with and how we plan to explore them. Then, we introduce our Datasets in section 3.2, followed by the intervention algorithms split based on the type of intervention in section 3.3 before finally delving into our evaluation metrics in section 3.4, final details about the implementation in section 3.5, and details about our generative AI usage in section 3.6.

3.1. Problem statement

Based on our background research detailed in chapter 2, we found that many studies fail to adequately address the bias and fairness considerations of proposed solutions. Typically, work focuses on maximizing accuracy on biased datasets such as MovieLens-1M, but this does not accurately reflect a model's performance, as research has shown that evaluating on MNAR datasets is unreliable due to the inherent bias in the dataset-creation process [33, 6]. To address this, some papers train novel solutions on MNAR data and evaluate them on MAR sets, providing a better sense of the model's true performance. However, this is predominantly done for debiasing solutions, as there is a distinction between debiasing tools and fairness-aware algorithms in the RecSys world. Research shows bias as built into the pipeline and an accuracy penalty, and fairness as mostly an external consideration, so the separation is understandable. However, there is considerable overlap between the two, with many fairness solutions, such as Zhu et al. [55], being based on notions of bias with an additional layer of equity or equality being applied based on a specific measurement of the recommendations.

The majority of the data we see published on new solutions, whether fairness or debiasing, is largely about their accuracy, with many papers either using custom bias/fairness metrics to estimate performance or flat-out omitting them, for example Wei et al. [44] and Zhang et al. [50]. The lack of such information makes it difficult for system designers to make an informed decision about which interventions suit them best, and it requires downstream researchers to manually estimate the bias and fairness outcomes of new solutions. Another aspect that RS research rarely considers is the impact interventions have on users. While we can find user-based studies in papers, they often focus on the entire user population, which muddies the results because interventions can affect users differently depending on how many popular items they interact with. For example, a user who primarily interacts with popular items might receive fewer relevant, i.e., unpopular, items due to an intervention algorithm that directly reduces the number of popular items in the final recommendation list, thereby hurting their satisfaction with the overall system.

Because of this, we are curious to see whether evaluating fairness-aware algorithms on MAR data will increase accuracy, as debiased algorithms do on debiased data. Additionally, we would like to investigate the fairness-outcome of debiasing solutions to learn to what degree they can correct unfair recommendations. Finally, while we can often find population-wide statistics on a recommender's performance (i.e., how the model performs across the entire test/validation set of items), we rarely see performance breakdowns by the solutions' impact on users. As such, we are curious to learn the impact

on different user groups, split by the ratio of popular items they interact with.

3.2. Datasets

Table 3.1: Dataset information

Dataset	Interactions	Users	Items	Head items	Tail items	Sparsity
MovieLens-1m	1,000,209	6041	3707	154	2699	95.53%
Amazon Software	1,276,840	146,397	17,592	90	15,745	99.950%
Yahoo	365,704	15,401	1001	32	569	97.62%
Coat	11,600	291	301	44	108	86.76%

In literature, we observe that approaches for correcting unfairness, such as `FA*IR`, exhibit lower accuracy. Intuitively, this makes sense, as fairness is a non-algorithmic decision made by the system designers. Often, fairness-aware pipelines aim to boost less-privileged items to achieve a more equitable distribution of options, but this may conflict with the user's preferences. At the same time, we observe the same phenomenon when testing debiasing tools on biased data, but not when testing against debiased data.

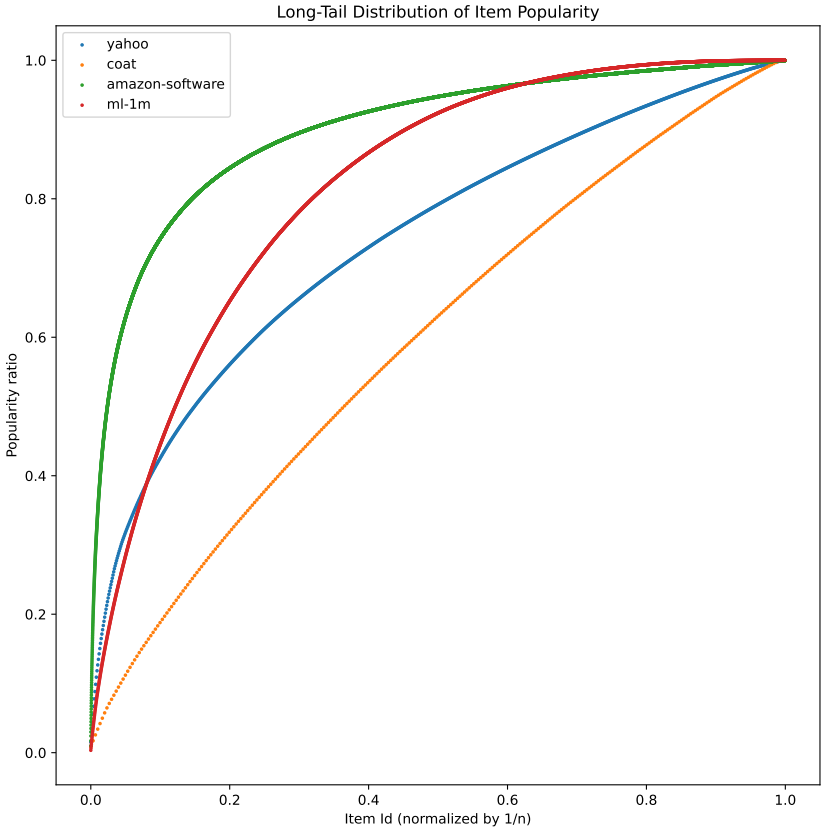


Figure 3.1: Normalized item popularity distribution

As such, we used both MNAR and MAR datasets in our experiments to observe whether the fairness-aware algorithms exhibit decreased performance on debiased data. Specifically, we consider MovieLens-1m [10], YahooR3 [19], CoatShopping [33], and Amazon-Software [12]. MovieLens-1m (ML-1m) is a widely used dataset in the Recommender Systems field due to its ease of availability and high quality, often serving as a benchmark. This dataset makes it easier to compare our work with other papers. However, ML-1m is an MNAR dataset and has been shown to have biases in previous work [33, 6].

On the other hand, YahooR3 and CoatShopping consist of both user-selected and randomly selected items, providing an MAR data split for experimentation. Specifically, in our case, we use the RecBole formulation of YahooR3 [53], which includes 25% of the MAR ratings in the training set, splitting the remaining 75% between the validation set (25%) and the testing set (50%). While our pipelines will be trained on MNAR data, the presence of MAR validation and test data allows us to get a realistic view of the model's performance. We could not find an existing RecBole version of CoatShopping, as such we processed the dataset into the required format. To aid the robustness of our experiments and make our conclusions better, we also include Amazon-Software, which is also MNAR. While we were able to find a RecBole formulation for Amazon-Software, we ultimately chose to use the publicly available pre-split dataset and process each split individually, as we did for CoatShopping. We did this because the RecBole version does not use the existing splits, making comparison to papers that use the original version harder.

3.3. Algorithms

For our models, we use two Collaborative Filtering formulations, Matrix Factorization and BPR. Matrix Factorization is a classic Collaborative Filtering model. In our case, we use Mean Squared Error (MSE) to train the model, which measures the average squared difference between the predicted ratings and the observed ratings in the dataset.

We also use the BPR [29] model, which uses the Bayesian Personalized Ranking loss:

$$\mathcal{L}_{BPR} = -\sum_{(u,i,j)\in D_s} \ln \sigma(\hat{x}_{ui} - \hat{x}_{uj}) \tag{3.1}$$

Unlike MF, which is a pointwise model, BPR is a pairwise model. The authors of [29] identified that prior methods trained on pointwise scores treat unobserved items as negative, which does not work for implicit feedback. This observation, along with others, allowed them to formulate a pairwise model with a custom loss function better suited to implicit feedback.

We choose to focus on Inverse Propensity Scoring (`IPS`), Model-agnostic Counterfactual Reasoning (`MACR`), Popularity Deconfounding and Adjustment (`PDA`), Popularity Compensation (`PC`), Pearson's correlation coefficient regularization (`PCC`), and `FA*IR` as our interventions. `IPS` and `FA*IR` have seen wide use in research due to their ease of use and the publicly available code for `FA*IR`. `MACR`, `PDA`, `PC`, and `PCC` are newer, state-of-the-art algorithms that will hopefully give us a better understanding of the current landscape. Furthermore, we found RecBole implementations for `IPS`, `MACR`, and `PDA`. We took those implementations and abstracted them to work for both of our models.

3.3.1. Debiasing

IPS

Inverse Propensity Scoring [33], also sometimes known as Inverse Propensity Weighting, is a debiasing intervention approach that utilizes propensities, defined as the conditional probability of assignment to a particular interaction given a vector of observed covariates [30], to adjust recommender scores. Specifically, under `IPS`, we multiply the pipeline's loss by the inverse of the propensities as follows:

$$\mathcal{L}' = \frac{1}{\mathcal{P}} * \mathcal{L} \tag{3.2}$$

Where $\mathcal{L}$ is the original loss, $\mathcal{P}$ are the propensities, and $\mathcal{L}$' is the adjusted loss. By scaling the loss by the inverse of the propensities, we ensure that more attention is paid to unpopular items that tend to have low propensity values.

Unfortunately, due to the nature of Recommender data, `IPS` can be difficult to apply. `IPS`-based solutions rely heavily on the propensities provided during training and inference. At the same time, it is difficult to estimate propensities from recommender data because recommender datasets are inherently dirty (MNAR) [33, 49, 31]. In particular, because some ratings are missing due to user preferences, it becomes incredibly difficult to calculate item propensities, as the data is inherently flawed and provides a biased view of item popularity. As such, we typically have to estimate the propensities, making

IPS highly sensitive to the estimation approach. Additionally, different estimation approaches exploit different biases, making the selection process more difficult, as designers need to first understand the biases the dataset and model might exhibit before settling on a pipeline. Furthermore, research has shown that propensity estimation approaches work best on MAR data [49, 31].

MACR
Model-agnostic counterfactual reasoning (MACR) [44] is one of many new approaches that leverage causal inference for debiasing. Specifically, the authors of MACR derived a causal inference equation that enabled them to formulate a model that absorbs bias via additional item and user modules. In particular, the item module estimates the influence of item popularity on a given item, while the user-item module predicts how likely a user is to interact with items. The authors then formulate this into a custom scoring equation:

$$\hat{y}_{ui} = \hat{y}_k * \sigma(\hat{y_i}) * \sigma(\hat{y_u}) \tag{3.3}$$

Where $\hat{y_i}$ and $\hat{y_u}$ are the outputs of the item and user modules, respectively. The authors then use individual loss functions for the additional modules alongside the main recommender's loss, and combine them using independent scaling factors for the item and user module losses. Finally, based on their causal derivation, the authors calculate the final ranking scores based on the following inference equation:

$$\hat{y}_{ui} = \hat{y}_k * \sigma(\hat{y}_i) * \sigma(\hat{y}_u) - c * \sigma(\hat{y}_i) * \sigma(\hat{y}_u) \tag{3.4}$$

Where c is a hyperparameter that directly controls the degree to which scores are adjusted. c scales the $[0 \dots 1]$ values outputted by the sigmoid, increasing the strength of the intervention at higher values. The authors show that increasing c provides a consistent benefit up to a point, after which performance degrades.

PDA
PDA [50] is another causal intervention approach that aims to leverage popularity bias to improve recommendation accuracy while mitigating its negative impact. Specifically, the authors introduce two models, PD and PDA. PD disentangles popularity bias from recommendations by applying a custom ELU function to the model's outputs, ensuring monotonicity. While PDA additionally adjusts the scores by scaling them with estimated popularity values as follows:

$$PDA_{ui} = ELU'(f_\theta(u, i)) \times (\tilde{m}_i)^{\tilde{\gamma}} \tag{3.5}$$

Where $\tilde{m}_i$ is the estimated popularity value and $\tilde{\gamma}$ is a smoothing factor. In the original paper, the authors do not spend much time on the popularity-estimation approach; instead, they use a simple time-series forecasting method to estimate the value. While not necessarily a problem, as we have seen from IPS, a popularity and propensity-based approach can be sensitive to the estimation approach used. As a result, the paper omits valuable information about PDA's behavior across different estimation techniques.

3.3.2. Fairness

FA*IR
FA*IR [48] is a popular post-processing reranking approach that has seen wide adoption due to its simplicity and ease of use, as the authors publicly host several versions of the algorithm on their GitHub. At its core, FA*IR greedily produces a fair ranking based on the hyperparameters it is provided. Notably, FA*IR uses an adjusted significance level to determine the minimum number of protected items required at each position, thereby allowing the authors to meet their group fairness condition, which states that the proportion of protected candidates must remain at or above the minimum p. This condition helps FA*IR better match user behavior, as users rarely scan the entire recommendation list; instead, they look only at the first few items.

FA*IR requires a protected class to determine which items to boost in the recommendation list. The authors originally defined this around the gender attribute of a particular dataset. However, since our intention is to review FA*IR's behavior across different datasets, we cannot select a specific user feature, as not all datasets may provide it. To this end, we define our protected set based on item popularity. Specifically, we treat the tail set as the protected set, providing an easy way to segregate items into protected and unprotected sets across all datasets.

PC
Popularity Compensation (PC) is a popularity-opportunity-based intervention that compensates Recommender scores for popularity-opportunity bias after recommendations. In particular, the authors set guidelines about how the compensation should behave. Specifically, compensation should align with item popularity and user preferences, while also being mindful of each user's value scale. To this effect, the authors first calculate the norm of the predicted scores:

$$n_u = \left|\left| \frac{(\hat{\mathbf{R}}_{u,:} \odot (1 - \mathbf{R}_{u,:}))}{(M - |O_u^+|} \right|\right|_F \tag{3.6}$$

They then calculate the popularity compensation score such that they meet their guidelines.

$$c_{u,i} = \frac{1}{pop(i)} \cdot (\hat{\mathbf{R}}_{u,i} \cdot \beta + 1 - \beta) \tag{3.7}$$

Here $\frac{1}{pop(i)}$ ensures the compensation follows item popularity, as $pop(i)$ returns the popularity of a given item, while $(\hat{\mathbf{R}}_{u,i} \cdot \beta + 1 - \beta)$ follows user preferences by treating the predicted score as an indicator for user preference and β is a trade-off weight to control the strength of the predicted scores in the final compensation.

$$\hat{\mathbf{R}}^*_{u,i} = \hat{\mathbf{R}}_{u,i} + \alpha \cdot C_{u,i} \cdot \frac{n_u}{m_u} \tag{3.8}$$

Finally, to maintain each user's value scale, the authors additionally normalize the scores using the norm of the compensation scores for items not in the training set.

PCC
Similar to PC, PCC is also a popularity-opportunity-based intervention. However, unlike PC, PCC is an in-processing regularization approach. Unlike its post-processing counterpart, PCC uses the Pearson correlation coefficient between the predicted scores for positive user-item pairs and the item popularity. PCC, like other regularization approaches, is sensitive to the regularization weight. Furthermore, the authors note that this regularization approach might negatively affect recommendation utility by complicating the model's task, as item popularity is continuous and unevenly distributed, making it difficult to minimize correlations.

3.4. Evaluation

To conduct our analysis, we first selected metrics to evaluate the accuracy, fairness, and bias of each pipeline. In this section, we discuss each metric we collect, including variations specific to this work, the justification for our choice of each metric, and the corresponding mathematical formulations.

3.4.1. Accuracy

To measure the impact of each intervention on accuracy, we chose nDCG and Recall, as both metrics have been widely adopted in the research literature, enabling us to more easily compare our results with other work. In particular, nDCG is a normalized variant of DCG that measures the effectiveness of recommendation results using a graded relevance scale. We can calculate nDCG using the following:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{3.9}$$

Where DCG_p and $IDCG_p$ are defined as follows:

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i+1)} \tag{3.10}$$

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{3.11}$$

And REL_p is the list of relevant documents. Unlike traditional `DCG`, `nDCG` is normalized to 0-1, making it easier to compare performance across runs.

Recall, on the other hand, is one of the simplest ML metrics. Based on relevance, Recall measures how often relevant items are retrieved by counting True Positives and False Negatives. Or in simpler terms:

$$Recall = \frac{TP}{TP + FN} \tag{3.12}$$

Where TP is the number of True Positives (i.e., relevant retrieved items), and FN is the number of relevant items not in the recommendation list. Both of these metrics give us valuable insight into the pipeline's accuracy. `nDCG` reports accuracy from a perspective that is better informed about user behavior, while Recall indicates how many relevant items are actually being recommended to users.

3.4.2. Bias

We reviewed several metrics that aim to quantify bias in Recommender systems, as discussed in chapter 2. Ultimately, we chose to use head-nDCG (`h-nDCG`) and tail-nDCG (`t-nDCG`), which allow us to see the `nDCG` for the most popular and least popular items, respectively. To elaborate, we constructed head and tail sets that each represent the top and bottom 25% of item popularity. We did this by first calculating each item's share of popularity by dividing each item's occurrences in the dataset by the total number of interactions. Then, starting from the most popular item for the headset and the least popular item for the tail set, we added items until each set's total share of popularity was about 25%. Unlike other approaches that sort by popularity and then directly take 25% of the total items, our approach ensures that our sets make up 25% of the total popularity, giving us a better sense of the accuracy of these items. Finally, we use these sets to filter out items before calculating `nDCG` as usual.

3.4.3. Fairness

For fairness, we settled on the `Gini Index` and `Item Coverage`. `Item Coverage` shows us the proportion of items included in the recommendation, allowing us to see how much of the catalog is being exposed to users. The `Gini Index` measures the inequality of a distribution. It is traditionally used as a measure of income inequality. However, research has shown [38] that the `Gini Index` can also serve as a useful recommendation metric. Essentially, the `Gini Index` measures the degree of variation in users' recommendation lists. The closer the Gini index is to 0, the more equally items are recommended across users. We specifically use the `Gini Index` definition described by RecBole:

$$GiniIndex@K = \frac{\sum_{i=1}^{|I|}(2i - |I| - 1)P(i)}{|I| \sum_{i=1}^{|I|} P(i)} \tag{3.13}$$

Where $P(i)$ is the number of times all items appear in the recommendation list.

3.5. Implementation details

We use the publicly available RecBole framework [46, 51, 52] alongside separate implementations for solutions not already in RecBole, as well as modifications to the framework. Our modifications and additional implementations are described in [25], and our experiment code is available at [24]. Specifically, we implemented `PDA`, `MACR`, `IPS`, `PC`, and `PCC`. `PDA`, `MACR`, and `IPS` have RecBole implementations

as defined in [53], but these implementations focus on a specific model backend, which does not fit our use case. Instead, we implemented extendable versions of each algorithm for the two models we use, `BPR` and `MF`. Additionally, we use the publicly available implementation for `FA*IR`. All of our experiments were run on DelftBlue [1]. We chose RecBole for its extensive support for various models and datasets, ease of use, and to ensure the reproducibility of our experiments. Furthermore, by extending the RecBole framework, we hope to enable work on the interaction between pipelines and bias- and fairness-intervention.

We tuned hyperparameters for all intervention approaches and models. Our results consist largely of data points from tuned algorithms. In some cases, specifically where intervention performance is directly controlled by a hyperparameter, we include data points for different tuning configurations so we can get a better idea of the algorithm's behavior.

3.6. Generative AI usage

We used Grammarly for writing assistance, primarily to fix spelling mistakes, punctuation errors, and poor phrasing. We wrote chapters inside Grammarly's online platform first and iteratively edited them using the provided suggestions. Upon completion, we moved each chapter into LaTeX and formatted the text together. We took full advantage of Grammarly's features thanks to the EDU license provided by TU Delft. We found the documentation provided by RecBole to be lacking in some cases, specifically regarding compatibility issues and internal functionality required to implement the intervention approaches. As such, we used generative AI tools to understand how RecBole works. Finally, we used generative AI to debug PyTorch implementation issues. In particular, we used generative AI to understand PyTorch error messages and CUDA compatibility issues on DelftBlue and local machines.

4

Results

In this chapter, we present the empirical results of our study. In section 4.1, we cover the first research question by comparing the performance of fairness-aware algorithms across two datasets. Then, in section 4.2, we compare the fairness outcomes of debiasing solutions, and, finally, in subsection 4.3.1, we discuss the user-group breakdown for each intervention to better understand which users are most affected.

As we discussed in section 3.5, all of our experiments were implemented inside and trained using RecBole. Before starting, we would like to address the large discrepancy between the results for ML-1m and YahooR3. In particular, results for YahooR3 are significantly lower than those for ML-1m, we expect this due to the sparsity of the YahooR3 dataset as discussed in section 3.2

4.1. Intervention impact on accuracy

The first question asks, ”What trends in accuracy can we observe when applying fairness-aware solutions on MAR data?”. To answer this question, we first compare the performance of the unintervened baselines on YahooR3 with that of the fairness-aware configurations. Specifically, we investigate the accuracy using nDCG and the debiasing accuracy using t-nDCG. We then repeat the comparison for CoatShopping, ML-1m, and Amazon-Software before comparing performance across datasets to identify clear trends. We also considered results comparing Recall to nDCG@20 but chose not to report them, as we found that, in general, Recall follows nDCG.

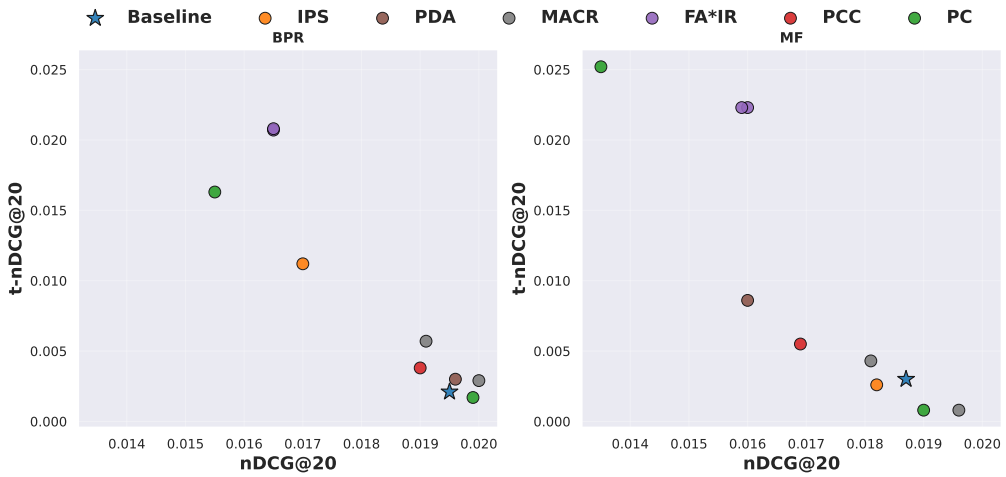


Figure 4.1: nDCG@20 vs t-nDCG@20 performance for YahooR3

As we can see in Figure 4.1, we see that PC, FA*IR, PCC, and MACR increase t-nDCG over the baseline, with FA*IR and PC showing the largest improvements. We should note that in cases where the hyperparameters directly control the intervention capacity (i.e., the ratio of protected items included in the final list, intervention scaling coefficients, etc.), such as FA*IR, PC, and MACR, we include data points for the best- and worst-performing configurations. As such, while both FA*IR configurations perform well, only the best-performing PC configuration beats the baseline. Overall, all interventions, except the worst-performing PC configuration, increase t-nDCG by at least a small amount for BPR. For MF, we see that more solutions reduce t-nDCG, with PC, IPS, and one MACR configuration all reducing t-nDCG. We see that some solutions that reduce t-nDCG also increase overall nDCG, as expected. t-nDCG measures the accuracy on the tail set of items, meaning that if we increase the amount of tail items in the recommendation list, t-nDCG should go up. If we reduce the number of tail items, we expect t-nDCG to decrease while nDCG increases. Interestingly, we observe contrasting results from IPS, possibly due to a mismatch in propensity. During experimentation, we tested different propensity estimation methods and found that BPR performs better with methods specifically designed for popularity bias, while MF performs better with rating-based methods. The poor result we are observing could be due to YahooR3 not having a strong rating bias, which limits the utility of IPS.

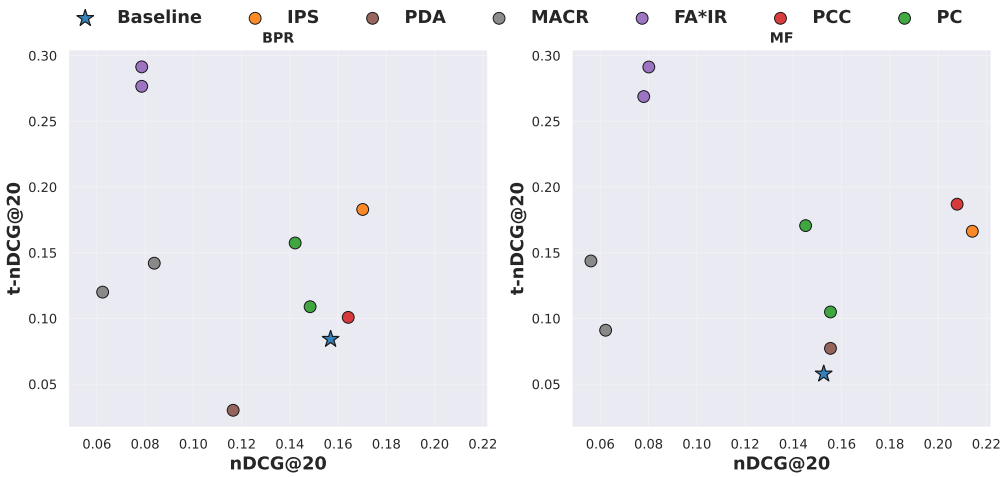


Figure 4.2: nDCG@20 vs t-nDCG@20 performance for CoatShopping

For CoatShopping, seen in Figure 4.1, we see much better results. Specifically, we see that only PDA reduces t-nDCG under BPR, and no intervention reduces it for MF. In contrast to YahooR3, we observe that PCC and IPS increase both the tail and overall nDCG for both models, supporting our choice of propensities for IPS. The reduction in t-nDCG observed for PDA under BPR could be due to its reliance on item popularity. PDA first aims to remove the effect of popularity bias, then reintroduces it into the ratings to help items that are popular due to their references. However, this also risks reintroducing a degree of popularity bias into the recommendation list, as we do not know where an item's popularity comes from. Simply put, an item can be popular because it is relevant, but it can also be popular because of exposure or selection bias, making the reintroduction of item popularity into the calculation dangerous.

In Figure 4.3, we see that PDA, FA*IR, and PCC are the only algorithms that consistently improve t-nDCG across models, with PC consistently underperforming regardless of model or configuration. At the same time, MACR shows improvement in the best configuration. PC uses item popularity in its calculations to adjust scores, while ML-1m is an MNAR dataset with dirty data. The poor performance we are witnessing may be due to the dirty test set used to calculate our metrics. In essence, the test set expects recommendations with a certain level of bias, which PC does not provide. The inherent bias in the testing set, combined with PDA's popularity-based adjustments, could also explain why this is the first dataset under which PDA performs better than the baseline for BPR.

Figure 4.4 showcases the results for Amazon-Software. We can see that PC performs much worse than in other datasets, even failing to improve t-nDCG over the baseline for BPR. The rest of the re-

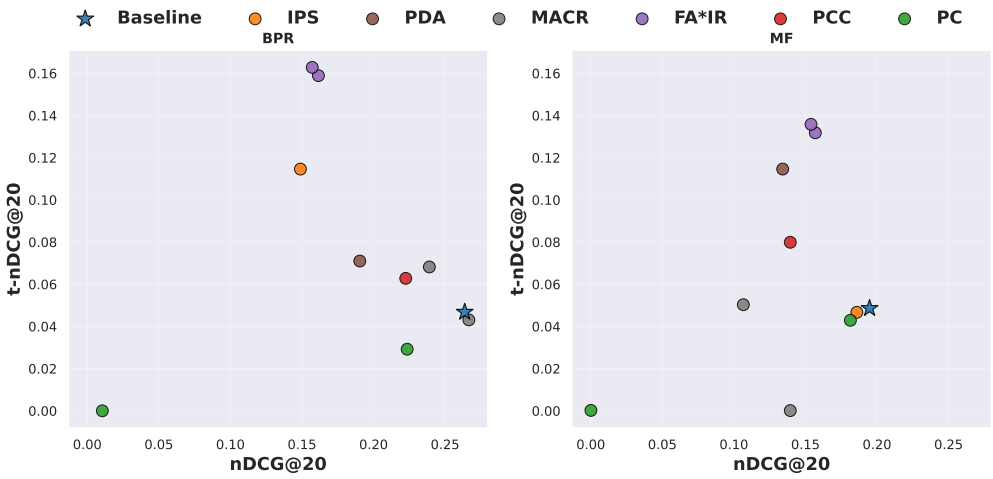


Figure 4.3: nDCG@20 vs t-nDCG@20 performance for ML-1m

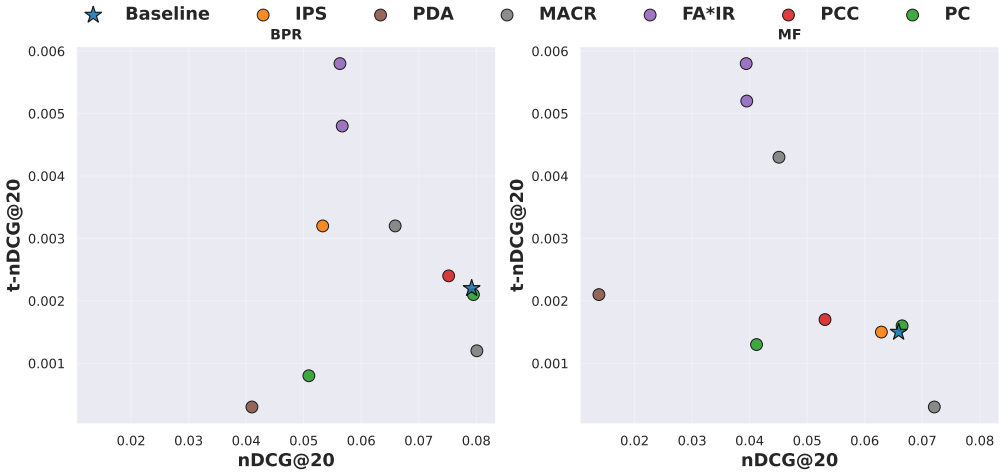


Figure 4.4: nDCG@20 vs t-nDCG@20 performance for Amazon-Software

sults are somewhat expected. Outside of ML-1m, PDA consistently underperforms on BPR, indicating a mismatch between PDA's intervention and BPR's loss function. We can expect PC and MACR configurations to perform worse as we include both the best- and worst-performing configurations. The tainted test set can easily explain the poor performance observed for PC, as we saw that PC performs well when tested against MAR test data.

With these results, we can clearly see that fairness-aware algorithms also benefit from MAR data, as several solutions struggle to perform well against MNAR data but show promising results against MAR test sets. While we do see some poor showings for interventions even under MAR data, in those cases, we can determine the issue is a mismatch between the intervention solution and the task at hand; for MNAR data, making this conclusion is much harder. Furthermore, evaluating on MNAR data would severely misrepresent algorithms such as PC, which performs better against MNAR data. However, we can also see that not all algorithms benefit from MNAR data, as both FA*IR and PCC consistently show improvements, though minor in some cases. Finally, we note the similar results seen between FA* IR configurations. Initially, we had believed YahooR3's results were so similar because of the dataset's sparsity, but Amazon-Software shows a much bigger difference and has a higher sparsity. Since the main difference between the configurations is the number of protected items in the final recommendation list, the contrast we observe may be due to a lack of niche items in the test data. FA*IR will always select protected items based on relevance, so after a certain number of protected

items, the only options left might have little to no interactions recorded in the test set.

4.2. Intervention impact on fairness

We conduct a similar analysis to section 4.1 for our second research question, ”What trends can we observe in the effects of debiasing RS optimizations on bias and fairness?”, which aims to evaluate the performance of debiasing interventions on bias and fairness. We first compare the performance of each debiasing intervention across models, then conduct a broader per-dataset comparison, starting with the MAR datasets. Furthermore, we should note that when selecting the best- and worst-performing results, we first filter all data points based on the best-performing metric that fits the given question: t-nDCG and nDCG for the first research question, and Gini Index and Item coverage for the second one. As a result, there might be differences in the results, since we might end up selecting different tuning targets.

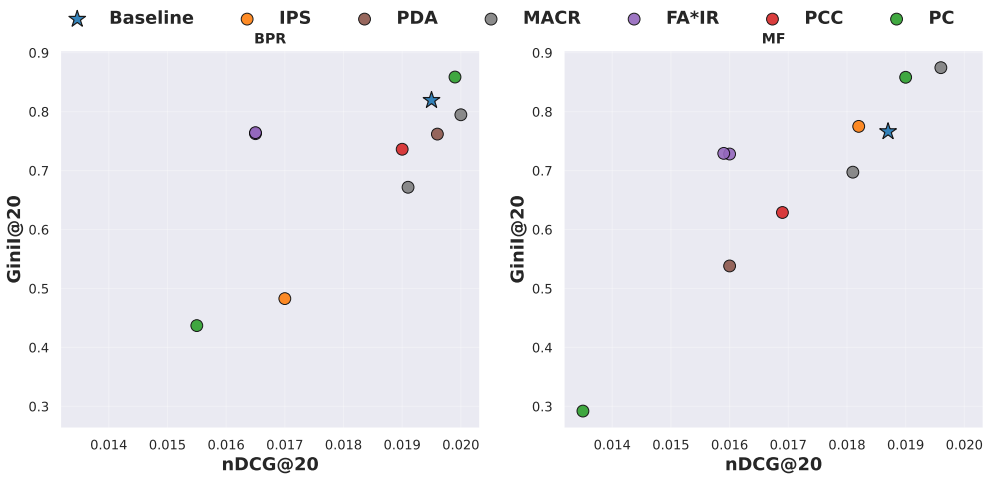


Figure 4.5: nDCG@20 vs Gini performance for YahooR3

In Figure 4.5, we see that almost all intervention solutions except PC and MACR (in their worst configurations) decrease the overall Gini index. Interestingly, we see that IPS shows similar performance to the baseline for MF, consistent with the results we discussed in **??**. We can see that the debiasing solutions provide some fairness benefits: IPS under BPR shows very strong results, and PDA does the same for MF, while MACR shows only minor improvements for both models.

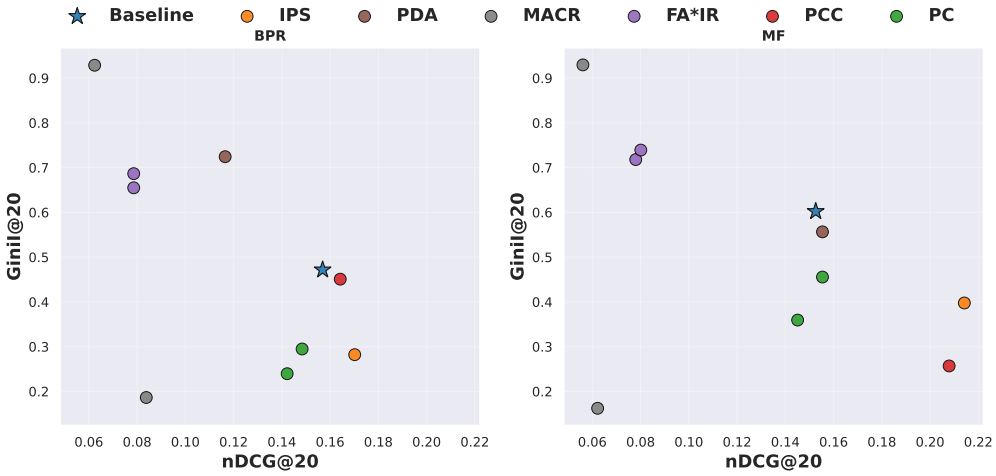


Figure 4.6: nDCG@20 vs Gini performance for CoatShopping

We can see more MAR results in Figure 4.6, where many solutions struggle to improve the Gini Index

relative to the baseline. Specifically, we see that FA*IR increases the Gini Index for both models. While this is unintuitive, this result is not that surprising given how we define our protected class. We define the protected class used by FA*IR as the 25% least popular items in the dataset; as such, FA*IR may be boosting a handful of niche items to all users, which would actually make recommendation lists more similar and could increase the Gini Index. The large difference we see in the MACR configurations can be explained by its hyperparameter c, which controls how much the original scores contribute to the final result, meaning that an incredibly high value might completely remove the model's scores and provide seemingly random recommendations to users, which would explain the low Gini Index. We also observe that PDA struggles to improve the Gini Index under BPR, possibly because it relies on item popularity.

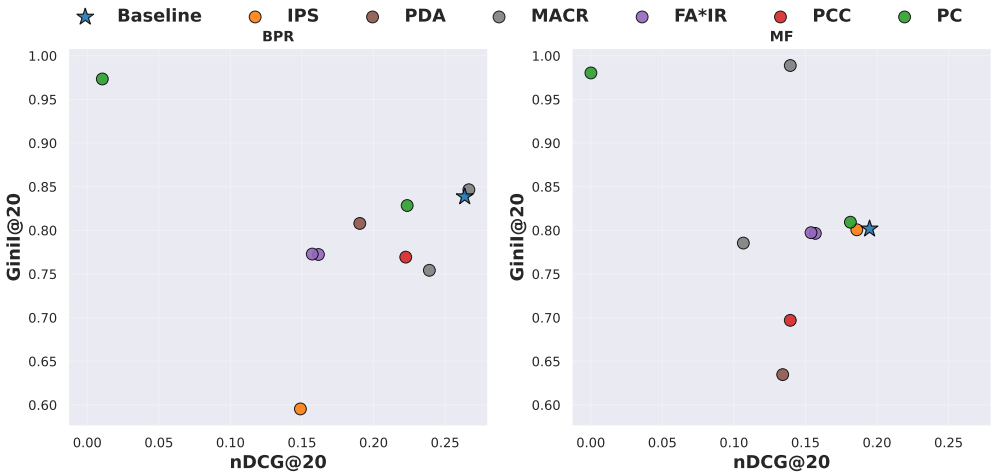


Figure 4.7: nDCG@20 vs Gini performance for ML-1m

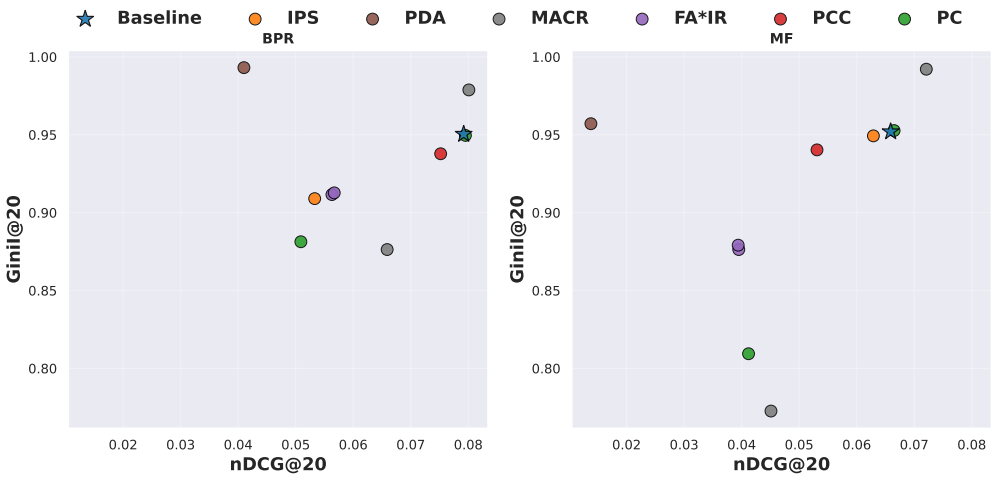


Figure 4.8: nDCG@20 vs Gini performance for Amazon-Software

For ML-1m, shown in Figure 4.7, we see that only the worst-performing PC and MACR configurations struggle to improve the Gini index for BPR, while PC fails outright for MF. Curiously, the benefit observed across all fairness solutions is rather limited, with many of them close to the baseline. In Figure 4.8, which shows the fairness results for Amazon-Software, we see improvements from IPS and MACR, and an increase in the Gini Index for PDA under BPR, whereas for MF, PDA, and IPS, both fail to improve on the baseline.

Broadly, we see that debiasing solutions improve fairness outcomes. However, the degree to which

they do that can be rather limited and likely depends on the dataset and model at hand. For example, we saw that PDA struggles to perform well on Amazon-Software, while it performs well on other datasets, but not on all models. We can observe this behavior in our results for the first research question as well. As a result, the benefit of these interventions depends on other parts of the pipeline.

4.3. Impact on user groups

4.3.1. User groups

In this section, we analyze user group results to answer our third research question: "What is the impact of bias and fairness intervention on user groups?" Specifically, first, we discuss how we evaluate impact across user groups, alongside our embedding space exploration analysis, and then we examine performance across five user groups. We segment the users based on the share of popular items in their interactions. For more details, please refer to subsection 4.3.1.

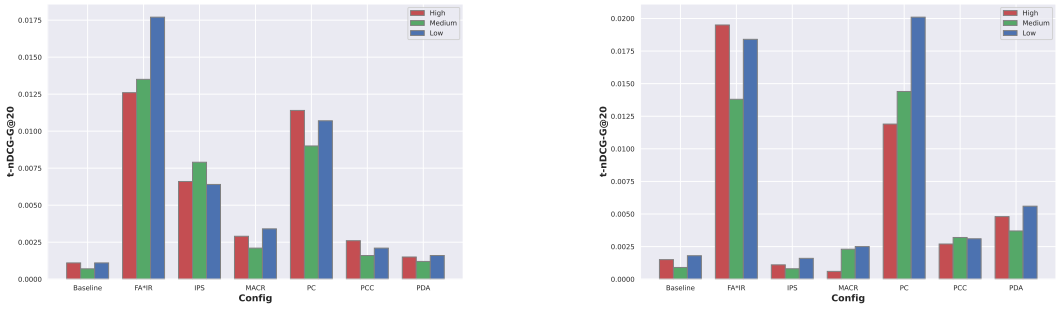
To aid our study on the impact of interventions across different user groups, we formulate grouped modifications to the previously discussed metrics. To be precise, we evenly divided the user set into predetermined groups based on the number of popular items they interact with. First, we used the h-nDCG headset to calculate the ratio of popular items each user interacts with. We then sorted the users and, similarly to the head and tail sets, evenly divided the items into five groups based on how many popular items each user interacted with. A concrete breakdown of the user groups is available in Table 4.1.

Dataset	High	Medium	Low
ML-1m	2579	1891	1572
Amazon-Software	25,115	35,289	85,994
Yahoo	4085	4504	6813
Coat	65	85	142

Table 4.1: User group counts by dataset

4.3.2. Embedding space exploration

To further understand the behavior of each intervention, we elect to construct embedding space plots for algorithms that directly modify the model's weights. This includes IPS, MACR, PDA, and PCC. Unfortunately, due to the embedding size, it is not possible to plot the embeddings directly. As such, we use Principal Component Analysis to reduce the embeddings' dimensionality to 3 and 2. We specifically collect plots in both 3D and 2D, but only report the 2D graphs. Additionally, we plot a user in the embedding space alongside 20 recommendations, with specific markings indicating whether each item appears in the user's interactions in the training and testing sets. Finally, we color-code all items by popularity to see how recommendations change as we apply interventions.



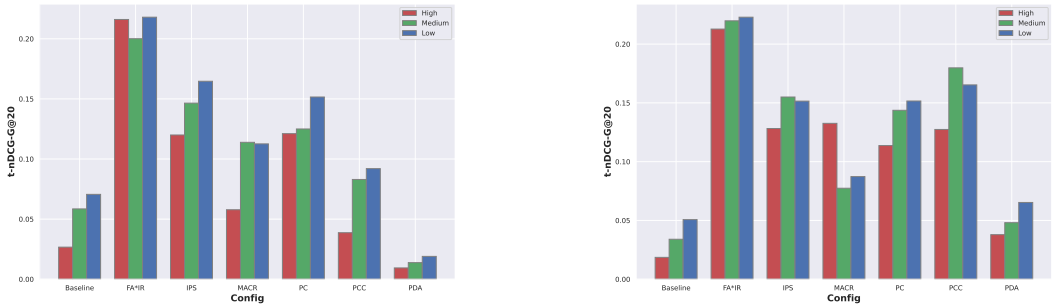
(a) BPR **(b)** MF

Figure 4.9: Yahoo user-group comparison results.

We can see more clearly that interventions impact different models differently in Figure 4.9. Specifically, in Figure 4.9a, we see that only FA*IR and IPS exhibit a different user-group impact distribution from the

others. Specifically, while most interventions show higher t-nDCG for the high and low groups, FA*IR shows the highest t-nDCG for the lowest group, followed by the middle group. In contrast, IPS shows the highest t-nDCG for the medium group, with the highest group close by. The result is interesting because we can see that while some solutions follow the baseline in their impact on a model, there is merit in investigating user-group performance directly.

Since t-nDCG measures accuracy on tail items, the higher performance for low-popularity users is due to the increased number of tail items in the final recommendation list. Interestingly, IPS, which should also increase the number of tail items, affects the middle group the most, suggesting that it better extracts user interests and includes relevant niche items in the final recommendations. For MF, as shown in Figure 4.9b, MACR and PC affect groups differently. To note, similarly to the previous two research questions, we have multiple data points for FA*IR, MACR, PC, and PDA. However, we chose to omit the lowest result in this section so we can compare solutions on a more even playing field and understand how interventions impact groups in a ”desirable” configuration. PC aims to reduce bias while also maintaining user interests. As such, PC may be promoting relevant tail items to all users, boosting t-nDCG across the board.

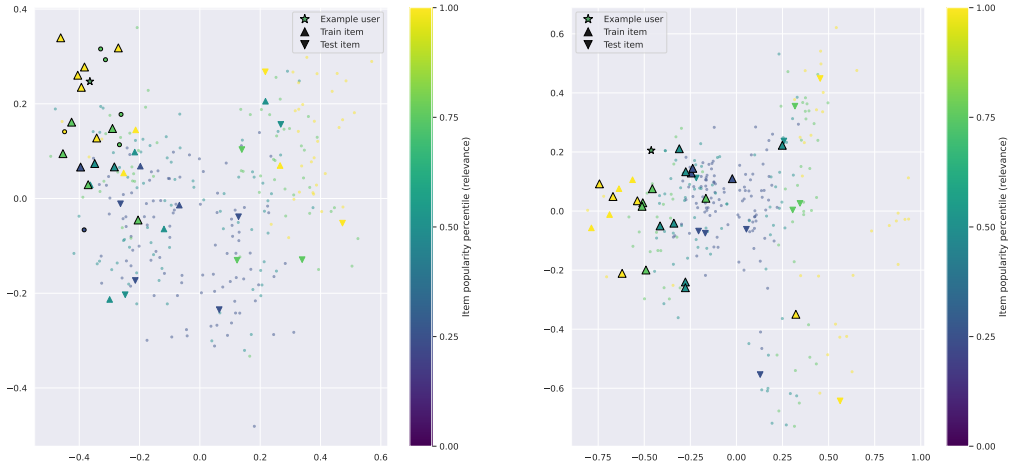


(a) BPR **(b)** MF

Figure 4.10: CoatShopping user-group comparison results

For CoatShopping, shown in Figure 4.10, we can again see that most configurations impact groups similarly, with FA*IR and MACR showing different distributions for BPR (Figure 4.10a). Specifically, FA*IR impacts the highest and lowest groups nearly equally, while MACR does the same for the medium and low groups. We do see cases where the distribution is uneven, i.e., one or more groups are affected less than the rest. However, generally, this still appears to follow the baseline distribution. When it comes to the MF results, seen in Figure 4.10b, we see more groups break away from the baseline in IPS, MACR, and PCC, all in different ways. IPS seems to, similarly to the BPR YahooR3 performance, primarily impact the middle group, with the lowest being a close second. Again, we believe these points demonstrate IPS's ability to support users' interest in niche items based on their propensities. We also see a similar pattern in PCC, with the medium and lowest groups showing the highest values, which supports the author's original observation that Pearson's correlation coefficient is related to item popularity. Specifically, a good showing in the middle group tells us that users with more varied opinions (both popular and unpopular item interests) are receiving better recommendations. Essentially, these users likely contain more niche items in their past interactions, and as such, increasing the number of unpopular items in recommendation lists would also increase this group's accuracy.

We show embedding plots of PCC on MF for the highest- and middle-popularity groups in CoatShopping to better understand this phenomenon. In Figure 4.11, which shows the baseline and PCC embedding plots for a randomly selected user from the middle group, we can actually see that the difference in performance is likely due to overfitting caused by PCC's regularization, causing the model to regurgitate the user's past interactions instead of exposing them to novel items. This observation means that, while at first glance PCC appears to show promising performance, in reality, it is simply learning from the user's past interactions, and any benefit we observe is due only to the user's original item interaction distribution. We believe that this supports our decision to use embedding plots in the user-group analysis, as it exposes the true cause of PCC's performance. Meaning that, had we reported only di-

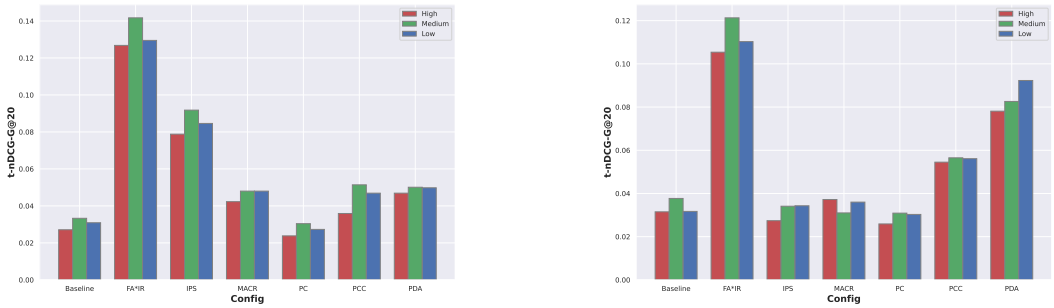


(a) Embedding space plot for the baseline.

(b) Embedding space plot for a the PCC.

Figure 4.11: Embedding space plots for Matrix Factorization on CoatShopping. The user was selected randomly from the middle group.

rect performance comparisons between configurations, we would have misrepresented PCC's actual performance. We could have confused future readers and system designers about the algorithm's benefits.



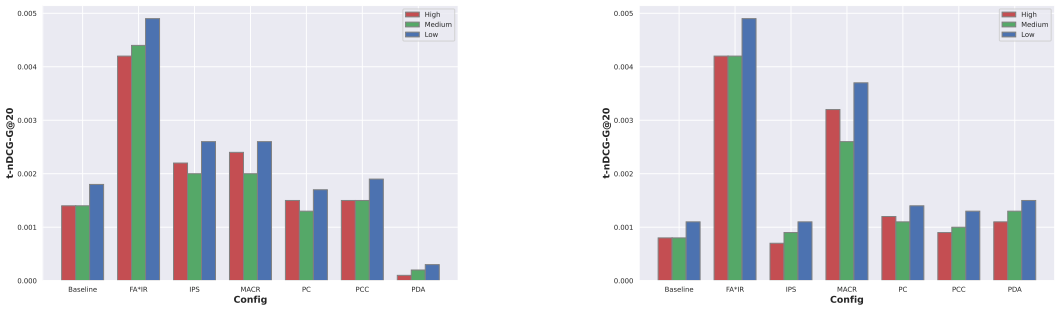
(a) BPR

(b) MF

Figure 4.12: MovieLens-1m user-group comparison results

Turning to the MNAR data, we see much greater consistency across user groups in MovieLens-1m, as shown in **??**, even across models. Specifically, we can see that for BPR, almost all configurations primarily impact the middle group, with MACR being the only exception, showing similar performance for the middle and lowest-popularity groups. We continue to see this trend in MF, except for MACR, which affects the highest and lowest groups most, and PDA, which affects the lowest and middle groups. While this result initially shows that interventions are more consistent for ML-1m, we must remember that we calculated these results against an MNAR test set. Hence, they are not truly representative of the model's performance. Again, we can expect the middle group to see larger benefits than the other groups, as they are likely to have more varied interests to exploit. However, the possibility that biases taint these interests casts doubt on the universality of these interventions.

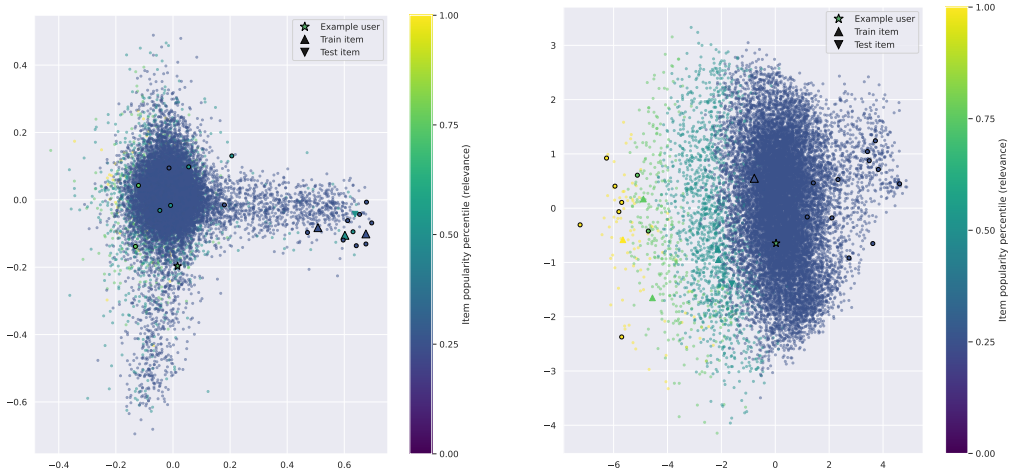
Unlike the ML-1m results, we actually see greater diversity in the distribution of impact across user groups in the Amazon-Software results, as shown in Figure 4.13. Specifically, we observe differing performance between FA*IR and PDA for BPR, with PDA performing significantly worse than the baseline. For MF, we see that the baseline and FA*IR show a similar trend, while IPS, PCC, and PDA are more similar to each other, and MACR is similar to PC. This observation means that our previous conclusion that MNAR data might yield more consistent results is incorrect. However, this does not diminish the fact that MNAR data remains tainted, and the performance we are witnessing is not realistic. It simply



(a) BPR **(b)** MF

Figure 4.13: Amazon-Software user-group comparison results

shows that there is a certain quality about ML-1m that makes interventions more consistent, possibly meaning that future work should consider other baselines for performance, as newer datasets, such as Amazon-Software, might provide more information about the behavior of individual interventions.



(a) Embedding space plot for the baseline. **(b)** Embedding space plot for a the PCC.

Figure 4.14: Embedding space plots for Matrix Factorization on CoatShopping. The user was selected randomly from the middle group.

Delving deeper into the embedding plots for MF on Amazon-Software in Figure 4.14, we can also see where the large gains in Gini Index discussed in section 4.2 come from. Specifically, looking at the baseline in Figure 4.14a, we can see a handful of repeated training items, while for the best-performing MACR configuration shown in Figure 4.14, we can see that the recommendations are almost exclusively novel to the user, with only one past interaction. Furthermore, we see a mix of niche and popular items. We should note, however, that this specific MACR configuration used a c value of 50, which goes past the $[0, 29]$ range discussed in the paper. In MACR, the c hyperparameter is subtracted directly from the model backbone scores and then scaled by the user and item modules, meaning it directly controls the degree to which we consider the original scores. Such a high value, coupled with promising results, might suggest that more complex model formulations that do not rely heavily on collaborative filtering scores might perform better than traditional models. During experimentation, we found that increasing the c value past the recommended range, while decreasing accuracy, can provide considerable benefits for debiasing and diversification. However, it appears that this benefit largely depends on the dataset, as not all datasets were as responsive to a higher c value.

There appears to be a significant benefit in conducting more fine-grained user-group-based studies and deeper analyses of the actual recommendation lists, either through embedding plots or other analytical approaches. We were able to more easily explain some of the performance we observed across our

results and to expose overfitting in one of the configurations. Overall, we learned that, while a solution might appear promising, we should be mindful of the driving force behind the improvement we observe. Furthermore, in some cases, the hyperparameter ranges reported in papers may not be sufficiently wide due to the complexity of recommender data. Simply put, a range that works well for one dataset may not adequately work for another.

5

Discussion

In this thesis, we investigate the performance of various intervention algorithms for bias and fairness across multiple benchmarks to determine whether there are consistent patterns in their behavior. Specifically, we experiment with BPR and MF trained on MovieLens-1M, YahooR3, CoatShopping, and Amazon-Software, and we evaluate 6 total interventions. Our research questions address the accuracy of fairness-aware algorithms on MAR data, the fairness-intervention capacity of debiasing algorithms across benchmarks, and their impact on different user groups. Our results, described in chapter 4, show a rather inconsistent picture. We do observe that some fairness interventions benefit from MAR data, although this effect is not consistent.

Furthermore, debiasing solutions can, in some cases, improve diversification outcomes, though this appears highly dependent on external factors such as the dataset, model, and hyperparameter configuration. Finally, we investigated the user group performance breakdown for each of our datasets, which allowed us to better understand that not all interventions impact users equally; further embedding-space analysis exposed glaring issues in at least one intervention approach.

5.1. Findings

Overall, our results paint an incomplete picture. While some algorithms appear to benefit from MNAR data, the benefit is not consistent across algorithms. We should also note that PCC was originally a debiasing solution. Still, we treat it as a fairness-aware algorithm because it applies the opportunity view to popularity bias. This decision casts some doubt on our results, as one can argue that the benefit we observe is due to PCC being a debiasing tool. However, PC, as described in the same paper, shows a different result. Meaning that, while some algorithms might not necessarily require MAR data to show good results, it is important to include such results to present a more informed picture of the performance of the proposed algorithm. We also observed a noticeable benefit to fairness outcomes from some debiasing solutions, reinforcing the connection between statistical bias and ethical bias. Finally, through our user group analysis, we identified possible issues with some of the interventions we discussed and clarified the cause of some results.

Furthermore, we observe that debiasing algorithms such as IPS, which we know benefit MAR data according to previous research, can still fail to exploit it fully. Designers need to be more careful about the specific conditions under which they apply algorithms. In particular, conducting exploratory studies to learn about specific qualities of the dataset, such as sparsity, which biases are prevalent, and how user groups interact with items, might be beneficial for system designers, as it would give them better clarity on which algorithms to use. Furthermore, we observe that some algorithms perform better for some models, even when the only difference is the loss function; this suggests that the interaction between a recommendation pipeline and intervention approaches is highly sensitive and warrants further study.

5.2. Limitations

One of, if not the biggest, limitation in our work is the scope. Specifically, we test only two models from the same family that differ only in their loss functions. Such a limited scope makes it very difficult to draw meaningful conclusions from our research questions. For all we know, we might observe better performance with a different model family, as they may be better suited to the interventions we discuss. Additionally, the intervention approaches we use are limited, as we predominantly employ in-processing and post-processing solutions, with no pre-processing algorithms. Furthermore, collecting more samples from both MAR and MNAR data could strengthen our results. However, this would be difficult due to the rarity of MAR data.

Furthermore, to apply every algorithm to all datasets, we had to assume certain formulations. Specifically, in the case of FA*IR, we define the protected class based on item popularity, which might negatively impact the algorithm's performance and explain why results are so close together. In its original formulation, FA*IR uses a protected class defined around an external user factor that is not related to relevance. However, in our case, as we have already discussed, item popularity is directly linked to the relevance learned by most recommendation models, meaning that we are essentially asking FA*IR to boost items that are inherently irrelevant to the model.

We experienced some difficulty implementing certain interventions in a reusable manner, as some of them were explicitly formulated around a specific kind of loss. For example, in the original paper, MACR is defined around BCE loss, which makes sense as the user and item modules each aim to estimate a probability distribution. However, this made it difficult to determine how to combine it with BPR properly. Initially, we applied BPR loss to all modules, but this formulation learned poorly and achieved very poor performance across the board. As such, we currently apply BPR loss only to the actual recommendation scores, with the item and user modules using BCE loss. We experienced additional complications due to the framework we chose to implement our algorithms within. While we recommend RecBole due to its ease of use and tools to support the research process, such as guarantees of reproducibility, we found the documentation somewhat lacking, making it much more difficult to figure out exactly where everything goes. We hope that the modifications we have made to the RecBole framework will make it much easier for researchers to carry out similar studies.

5.3. Future Work

Regarding future work, we recommend expanding the scope by including additional datasets and models, so researchers can investigate whether the patterns we observe are consistent across MAR datasets and across recommendation model types. Furthermore, we observe that the propensity estimation approach plays a large role in how well an intervention performs. As such, future work could directly explore the impact of various propensity estimations across a series of datasets and algorithms similar to what we have done here. Such a study provides valuable information on which propensities work best for which datasets and models, while uncovering hidden connections between the dataset/-model's underlying structure and the impact of the propensities. We already know that some models are more susceptible to certain biases, for example, BPR explicitly learns to rank positive (observed) items above negative (unobserved) ones. This behavior means that popular items, which are more likely to appear in user interactions, would inherently be ranked higher. Simply put, a popular item might appear as positive thousands of times, pushing its score higher and higher, while a tail item could only appear in a handful of examples.

6

Conclusion

Overall, it is difficult to draw a clear conclusion from our work. While we observe that some interventions benefit from MAR data, this is not consistent and may depend on the specific model. We observe improved diversification performance from debiasing solutions, although it may be limited. Regarding user groups, poor algorithmic intervention might lead to disparate impact across them. Intuitively, this makes sense, as certain users might be more inclined to interact with unpopular or diverse items because of the number of popular items they've historically interacted with. We hope that our contributions, in the form of our study and subsequent results, along with contributions to the RecBole framework, will foster further research on the behavior of intervention algorithms in RS.

Additionally, we hope that, through this work, we show the value of results from debiasing and fairness algorithms that use agnostic metrics, i.e., metrics that do not rely on a hyper-specific definition of bias or fairness. Such metrics allow us to better compare algorithms across papers and help system designers better understand the impact of the interventions they are considering. Finally, we hope that through our user group and embedding space analysis, we were able to show the merit of non-traditional (i.e., recommendation list metric) evaluation of recommender systems, as it allows us to understand better how modifications to the recommendation pipeline impact users, and allowed us to expose clear overfitting in one of our intervention approaches.

References

[1] Delft High Performance Computing Centre (DHPC). *DelftBlue Supercomputer (Phase 2)*. https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2. 2024.

[2] Aristotle. *The Nicomachean ethics*. Oxford University Press, USA, June 2009.

[3] Jeffrey J Arnett. "The neglected 95%: why American psychology needs to become less American". en. In: *Am Psychol* 63.7 (Oct. 2008), pp. 602–614.

[4] 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. ACM, June 2018, pp. 405–414. DOI: 10.1145/3209978.3210063. URL: http://dx.doi.org/10.1145/3209978.3210063.

[5] 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. ACM, 2018, pp. 405–414. DOI: 10.1145/3209978.3210063. URL: http://dx.doi.org/10.1145/3209978.3210063.

[6] Diego Carraro and Derek Bridge. "A sampling approach to Debiasing the offline evaluation of recommender systems". In: *Journal of Intelligent Information Systems* 58.2 (July 2021), pp. 311–336. DOI: 10.1007/s10844-021-00651-y. URL: https://doi.org/10.1007/s10844-021-00651-y.

[7] Jiawei Chen et al. "Bias and Debias in Recommender System: A Survey and Future Directions". In: *ACM Trans. Inf. Syst.* 41.3 (Feb. 2023). ISSN: 1046-8188. DOI: 10.1145/3564284. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3564284.

[8] Michael D. Ekstrand et al. "All The Cool Kids, How Do They Fit In?: Popularity and Demographic Biases in Recommender Evaluation and Effectiveness". In: *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. Ed. by Sorelle A. Friedler and Christo Wilson. Vol. 81. Proceedings of Machine Learning Research. PMLR, 23–24 Feb 2018, pp. 172–186. URL: https://proceedings.mlr.press/v81/ekstrand18b.html.

[9] Patrick Grother, Mei Ngan, and Kayee Hanaoka. *Face recognition vendor test part 3: demographic effects*. en. Tech. rep. NIST IR 8280. Gaithersburg, MD: National Institute of Standards and Technology, Dec. 2019, NIST IR 8280. DOI: 10.6028/NIST.IR.8280. URL: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8280.pdf (visited on 04/24/2026).

[10] F. Maxwell Harper and Joseph A. Konstan. "The MovieLens Datasets: History and Context". In: *ACM Trans. Interact. Intell. Syst.* 5.4 (Dec. 2015). ISSN: 2160-6455. DOI: 10.1145/2827872. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/2827872.

[11] Joseph Henrich, Steven J. Heine, and Ara Norenzayan. "The weirdest people in the world?" In: *Behavioral and Brain Sciences* 33.2–3 (2010), pp. 61–83. DOI: 10.1017/S0140525X0999152X.

[12] Yupeng Hou et al. "Bridging Language and Items for Retrieval and Recommendation". In: *arXiv preprint arXiv:2403.03952* (2024).

[13] Cheng-Kang Hsieh et al. "Collaborative Metric Learning". In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 193–201. ISBN: 9781450349130. DOI: 10.1145/3038912.3052639. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3038912.3052639.

[14] Jin Huang et al. “Going Beyond Popularity and Positivity Bias: Correcting for Multifactorial Bias in Recommender Systems”. In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’24. Washington DC, USA: Association for Computing Machinery, 2024, pp. 416–426. ISBN: 9798400704314. DOI: 10.1145/3626772.3657749. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3626772.3657749.

[15] Guido Imbens and Donald B. Rubin. *Causal inference for statistics, social, and biomedical sciences : an introduction*. eng. Cambridge: Cambridge University Press, 2015. ISBN: 9781139025751. URL: https://login.eux.idm.oclc.org/login?url=https://doi.org/10.1017/CBO9781139025751.

[16] Anastasiia Klimashevskaia et al. “A survey on popularity bias in recommender systems”. In: *User Modeling and User-Adapted Interaction* 34.5 (2024), pp. 1777–1834. ISSN: 1573-1391. DOI: 10.1007/s11257-024-09406-0. URL: http://dx.doi.org/10.1007/s11257-024-09406-0.

[17] Deepak Kumar et al. “Fairness of recommender systems in the recruitment domain: an analysis from technical and legal perspectives”. en. In: *Front Big Data* 6 (Oct. 2023), p. 1245198.

[18] Zachary C. Lipton and Jacob Steinhardt. *Troubling Trends in Machine Learning Scholarship*. 2018. arXiv: 1807.03341 [stat.ML]. URL: https://arxiv.org/abs/1807.03341.

[19] Benjamin M. Marlin and Richard S. Zemel. “Collaborative prediction and ranking with non-random missing data”. In: *Proceedings of the Third ACM Conference on Recommender Systems*. RecSys ’09. New York, New York, USA: Association for Computing Machinery, 2009, pp. 5–12. ISBN: 9781605584355. DOI: 10.1145/1639714.1639717. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/1639714.1639717.

[20] Christian H. Nolte, Peter U. Heuschmann, and Matthias Endres. “Sex and Gender Differences in Neurology”. In: *Sex and Gender Aspects in Clinical Medicine*. Ed. by Sabine Oertelt-Prigione and Vera Regitz-Zagrosek. London: Springer London, 2012, pp. 169–182. ISBN: 978-0-85729-832-4. DOI: 10.1007/978-0-85729-832-4_11. URL: https://doi.org/10.1007/978-0-85729-832-4_11.

[21] Zohreh Ovaisi et al. “Correcting for Selection Bias in Learning-to-rank Systems”. In: *Proceedings of The Web Conference 2020*. WWW ’20. ACM, Apr. 2020, pp. 1863–1873. DOI: 10.1145/3366423.3380255. URL: http://dx.doi.org/10.1145/3366423.3380255.

[22] Vincenzo Paparella et al. “Reproducibility of Multi-Objective Reinforcement Learning Recommendation: Interplay between Effectiveness and Beyond-Accuracy Perspectives”. In: *Proceedings of the 17th ACM Conference on Recommender Systems*. RecSys ’23. Singapore, Singapore: Association for Computing Machinery, 2023, pp. 467–478. ISBN: 9798400702419. DOI: 10.1145/3604915.3609493. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3604915.3609493.

[23] Gourab K Patro et al. “FairRec: Two-Sided Fairness for Personalized Recommendations in Two-Sided Platforms”. In: *Proceedings of The Web Conference 2020*. WWW ’20. Taipei, Taiwan: Association for Computing Machinery, 2020, pp. 1194–1204. ISBN: 9781450370233. DOI: 10.1145/3366423.3380196. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3366423.3380196.

[24] Petar Petrov. *Experiment framework*. Version 1.0.0.

[25] Petar Petrov and RecBole Team. *RecBole - Algorithmic Bias add-on*. Version 2.0.4.

[26] Bruno Pradel, Nicolas Usunier, and Patrick Gallinari. “Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics”. In: *Proceedings of the Sixth ACM Conference on Recommender Systems*. RecSys ’12. Dublin, Ireland: Association for Computing Machinery, 2012, pp. 147–154. ISBN: 9781450312707. DOI: 10.1145/2365952.2365982. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/2365952.2365982.

[27] M Rademaker. “Do women have more adverse drug reactions?” en. In: *Am J Clin Dermatol* 2.6 (2001), pp. 349–351.

[28] Bashir Rastegarpanah, Krishna P. Gummadi, and Mark Crovella. “Fighting Fire with Fire: Using Antidote Data to Improve Polarization and Fairness of Recommender Systems”. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. WSDM ’19. Melbourne VIC, Australia: Association for Computing Machinery, 2019, pp. 231–239. ISBN: 9781450359405. DOI: 10.1145/3289600.3291002. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3289600.3291002.

[29] Steffen Rendle et al. “BPR: Bayesian personalized ranking from implicit feedback”. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. UAI ’09. Montreal, Quebec, Canada: AUAI Press, 2009, pp. 452–461. ISBN: 9780974903958.

[30] Paul R. Rosenbaum and Donald B. Rubin. “The Central Role of the Propensity Score in Observational Studies for Causal Effects”. In: *Biometrika* 70.1 (1983), pp. 41–55. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/2335942 (visited on 04/24/2026).

[31] Yuta Saito and Masahiro Nomura. *Towards Resolving Propensity Contradiction in Offline Recommender Learning*. 2022. arXiv: 1910.07295 [stat.ML]. URL: https://arxiv.org/abs/1910.07295.

[32] Beatrice Savoldi et al. “A decade of gender bias in machine translation”. In: *Patterns* 6.6 (2025), p. 101257. ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2025.101257. URL: https://www.sciencedirect.com/science/article/pii/S2666389925001059.

[33] Tobias Schnabel et al. *Recommendations as Treatments: Debiasing Learning and Evaluation*. 2016. arXiv: 1602.05352 [cs.LG]. URL: https://arxiv.org/abs/1602.05352.

[34] Dimitris Serbos et al. “Fairness in Package-to-Group Recommendations”. In: *Proceedings of the 26th International Conference on World Wide Web*. WWW ’17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, pp. 371–379. ISBN: 9781450349130. DOI: 10.1145/3038912.3052612. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3038912.3052612.

[35] Yang Shi, Guannan Liang, and Young-joo Chung. *Meta-Shop: Improving Item Advertisement For Small Businesses*. 2022. arXiv: 2212.01414 [cs.IR]. URL: https://arxiv.org/abs/2212.01414.

[36] Dusan Stamenkovic et al. “Choosing the Best of Both Worlds: Diverse and Novel Recommendations through Multi-Objective Reinforcement Learning”. In: *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. WSDM ’22. Virtual Event, AZ, USA: Association for Computing Machinery, 2022, pp. 957–965. ISBN: 9781450391320. DOI: 10.1145/3488560.3498471. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3488560.3498471.

[37] Harald Steck. “Calibrated recommendations”. In: *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys ’18. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2018, pp. 154–162. ISBN: 9781450359016. DOI: 10.1145/3240323.3240372. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3240323.3240372.

[38] Wenlong Sun et al. “Debiasing the Human-Recommender System Feedback Loop in Collaborative Filtering”. In: *Companion Proceedings of The 2019 World Wide Web Conference*. WWW ’19. San Francisco, USA: Association for Computing Machinery, 2019, pp. 645–651. ISBN: 9781450366755. DOI: 10.1145/3308560.3317303. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3308560.3317303.

[39] Özge Sürer, Robin Burke, and Edward C. Malthouse. “Multistakeholder recommendation with provider constraints”. In: *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys ’18. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2018, pp. 54–62. ISBN: 9781450359016. DOI: 10.1145/3240323.3240350. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3240323.3240350.

[40] Douglas R. Turnbull et al. *Exploring Popularity Bias in Music Recommendation Models and Commercial Steaming Services*. 2022. arXiv: 2208.09517 [cs.IR]. URL: https://arxiv.org/abs/2208.09517.

[41] Daniel Valcarce et al. “On the robustness and discriminative power of information retrieval metrics for top-N recommendation”. In: *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys ’18. Vancouver, British Columbia, Canada: Association for Computing Machinery, 2018, pp. 260–268. ISBN: 9781450359016. DOI: 10.1145/3240323.3240347. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3240323.3240347.

[42] Margaret Waltz, Anne Drapkin Lyerly, and Jill A. Fisher. “Exclusion of Women from Phase I Trials: Perspectives from Investigators and Research Oversight Officials”. In: *Ethics & Human Research* 45.6 (2023), pp. 19–30. DOI: https://doi.org/10.1002/eahr.500170. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/eahr.500170. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/eahr.500170.

[43] Yifan Wang et al. “A Survey on the Fairness of Recommender Systems”. In: *ACM Transactions on Information Systems* 41.3 (Feb. 2023), pp. 1–43. ISSN: 1558-2868. DOI: 10.1145/3547333. URL: http://dx.doi.org/10.1145/3547333.

[44] Tianxin Wei et al. “Model-Agnostic Counterfactual Reasoning for Eliminating Popularity Bias in Recommender System”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. KDD ’21. Virtual Event, Singapore: Association for Computing Machinery, 2021, pp. 1791–1800. ISBN: 9781450383325. DOI: 10.1145/3447548.3467289. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3447548.3467289.

[45] Lin Xiao et al. “Fairness-Aware Group Recommendation with Pareto-Efficiency”. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys ’17. Como, Italy: Association for Computing Machinery, 2017, pp. 107–115. ISBN: 9781450346528. DOI: 10.1145/3109859.3109887. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3109859.3109887.

[46] Lanling Xu et al. “Towards a More User-Friendly and Easy-to-Use Benchmark Library for Recommender Systems”. In: *SIGIR*. ACM, 2023, pp. 2837–2847.

[47] Shuyuan Xu et al. *Causal Inference for Recommendation: Foundations, Methods and Applications*. 2023. arXiv: 2301.04016 [cs.IR]. URL: https://arxiv.org/abs/2301.04016.

[48] Meike Zehlike et al. “FA*IR: A Fair Top-k Ranking Algorithm”. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM ’17. ACM, Nov. 2017, pp. 1569–1578. DOI: 10.1145/3132847.3132938. URL: http://dx.doi.org/10.1145/3132847.3132938.

[49] Honglei Zhang et al. “Uncovering the Propensity Identification Problem in Debiased Recommendations”. In: *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 2024, pp. 653–666. DOI: 10.1109/ICDE60146.2024.00056.

[50] Yang Zhang et al. “Causal Intervention for Leveraging Popularity Bias in Recommendation”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 11–20. ISBN: 9781450380379. DOI: 10.1145/3404835.3462875. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3404835.3462875.

[51] Wayne Xin Zhao et al. “RecBole 2.0: Towards a More Up-to-Date Recommendation Library”. In: *CIKM*. ACM, 2022, pp. 4722–4726.

[52] Wayne Xin Zhao et al. “RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms”. In: *CIKM*. ACM, 2021, pp. 4653–4664.

[53] Wayne Xin Zhao et al. “Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms”. In: *CIKM*. 2021.

[54] Yu Zheng et al. *Disentangling User Interest and Conformity for Recommendation with Causal Embedding*. 2021. arXiv: 2006.11011 [cs.IR]. URL: https://arxiv.org/abs/2006.11011.

[55] Ziwei Zhu et al. “Popularity-Opportunity Bias in Collaborative Filtering”. In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. WSDM ’21. Virtual Event, Israel: Association for Computing Machinery, 2021, pp. 85–93. ISBN: 9781450382977. DOI: 10.1145/3437963.3441820. URL: https://doi-org.tudelft.idm.oclc.org/10.1145/3437963.3441820.