Bias and Debias in Recommender System: A Survey and Future Directions

JIAWEI CHEN and HANDE DONG, University of Science and Technology of China, China
XIANG WANG and FULI FENG, National University of Singapore, Singapore
MENG WANG, Hefei University of Technology, China
XIANGNAN HE[†], University of Science and Technology of China, China

While recent years have witnessed a rapid growth of research papers on recommender system (RS), most of the papers focus on inventing machine learning models to better fit user behavior data. However, user behavior data is observational rather than experimental. This makes various biases widely exist in the data, including but not limited to selection bias, position bias, exposure bias, and popularity bias. Blindly fitting the data without considering the inherent biases will result in many serious issues, e.g., the discrepancy between offline evaluation and online metrics, hurting user satisfaction and trust on the recommendation service, etc. To transform the large volume of research models into practical improvements, it is highly urgent to explore the impacts of the biases and perform debiasing when necessary. When reviewing the papers that consider biases in RS, we find that, to our surprise, the studies are rather fragmented and lack a systematic organization. The terminology "bias" is widely used in the literature, but its definition is usually vague and even inconsistent across papers. This motivates us to provide a systematic survey of existing work on RS biases. In this paper, we first summarize seven types of biases in recommendation, along with their definitions and characteristics. We then provide a taxonomy to position and organize the existing work on recommendation debiasing. Finally, we identify some open challenges and envision some future directions, with the hope of inspiring more research work on this important yet less investigated topic. The summary of debiasing methods reviewed in this survey can be found at https://github.com/jiawei-chen/RecDebiasing.

1 INTRODUCTION

Being able to provide personalized suggestions to each user, recommender system (RS) has been recognized as the most effective way to alleviate information overloading. It not only facilitates users seeking information, but also benefits content providers with more potentials of making profits. Nowadays, recommendation techniques have been intensively used in countless applications, e.g., E-commerce platforms (Alibaba, Amazon), social networks

[†]Corresponding author: hexn@ustc.edu.cn.
Authors' addresses: Jiawei Chen, cjwustc@ustc.edu.cn; Hande Dong, donghd@mail.ustc.edu.cn, University of Science and Technology of China, Hefei, China; Xiang Wang, xiangwang@u.nus.edu; Fuli Feng, fulifeng93@gmail.com, National University of Singapore, Singapore; Meng Wang, eric.mengwang@gmail.com, Hefei University of Technology, Hefei, China; Xiangnan He[†], hexn@ustc.edu.cn, University of Science and Technology of China, Hefei, China.

(Facebook, Weibo), video-sharing platforms (YouTube, TikTok), lifestyle apps (Yelp, Meituan), and so on. As such, the importance of RS cannot be overstated especially in the era that the information overload issue becomes increasingly serious.

Ubiquity of Biases in RS. Although RS has generated large impacts in a wide range of applications, it faces many bias problems which are challenging to handle and may deteriorate the recommendation effectiveness. Bias is common in RS for the following factors. (1) User behavior data, which lays the foundation for recommendation model training, is observational rather than experimental. The main reason is that a user generates behaviors on the basis of the exposed items, making the observational data confounded by the exposure mechanism of the system and the self-selection of the user. (2) Items are not evenly presented in the data, e.g., some items are more popular than others and thus receive more user behaviors. As a result, these popular items would have a larger impact on the model training, making the recommendations biased towards them. The same situation applies to the user side. (3) One nature of RS is the feedback loop — the exposure mechanism of the RS determines user behaviors, which are circled back as the training data for the RS. Such feedback loop not only creates biases but also intensifies biases over time, resulting in "the rich get richer" Matthew effect.

Increasing Importance of Biases in RS Research. Recent years have seen a surge of research effort on recommendation biases. Figure 1 shows the number of related papers in top venues increases significantly since the year of 2015. The prestigious international conference on information retrieval, SIGIR, has organized specific sessions in 2020 and 2021 to discuss topics on bias elimination[1]. SIGIR even presents the Best Paper award to the paper on this topic in 2018 [23], 2020 [121] and 2021 [124, 208], respectively. The conferences Recsys and WWW also organized tutorial on this topic in 2021 [33, 182]. Biases not only draw increasing attention from the information retrieval academia, but also from the industry. For example, one competing task of KDD Cup 2020 organized by Alibaba is to handle the long-tail bias in E-commerce recommendation[2].

Necessity of this Survey. Although many papers are published on this topic recently, to the best of our knowledge, none of them has provided a global picture of the RS biases and corresponding debiasing techniques. Particularly, we find that current studies on this topic are rather fragmented — despite the wide usage of the terminology "bias" in the literature, its definition is usually vague and even inconsistent across papers. For example, some work use "selection bias" to denote the bias of observed rating values [140], while others use "observational bias" to refer to the same meaning instead [66]. More confusingly, the same terminology "selection bias" has been conceptualized differently in different publications [125, 140, 169]. Moreover, a considerable number of researchers do not explicitly mention "bias" or "debias" in the paper (e.g. [29, 102, 174]), but they indeed address one type of biases in RS; these significant related work is difficult to be retrieved by the researchers interested in the bias topic. Given the increasing attention of biases in RS, the rapid development of debiasing techniques, and the flourishing but fragmented publications, we believe it is the right time to present a survey of this area, so as to benefit the successive researchers and practitioners to understand current progress and further work on this topic.

Difference with Existing Surveys. A number of surveys in recommendation have been published recently, focusing on different perspectives of RS. For example, [207] reviews explainable recommendation, [156] reviews knowledge-based recommendation, [203] and [209] summarize the recommendation methods based on deep learning and reinforcement learning, respectively. However, to our knowledge, the perspective of bias has not been reviewed in existing RS surveys. There are some surveys on the bias issues, but they are not on the recommendation task. For example, [16] recently reviews the bias issues in natural language processing, [159] reviews the sample selection bias on model estimation, [206] summarizes fairness in learning-based sequential decision algorithms. There are some surveys on the bias and fairness of general machine learning and artificial

---

[1]http://www.sigir.org/sigir2020/schedule/; https://sigir.org/sigir2021/schedule/
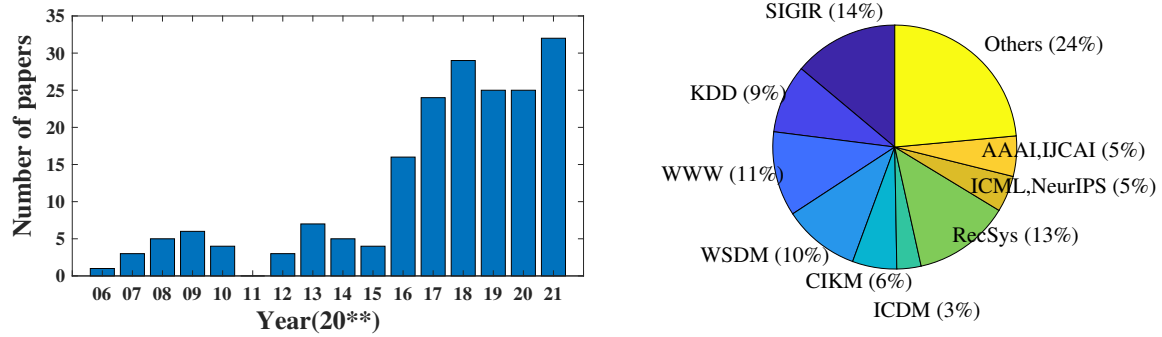[2]https://tianchi.aliyun.com/competition/entrance/231785/introduction

Fig. 1. The statistics of publications related to biases in RS with the publication year and venue.

intelligence systems [44, 118, 123, 145]. Comparing to the bias issues in other tasks, bias in RS has its own characteristics, requiring a new inclusive review and summary. To this end, we make the following contributions in this survey:

- Summarizing seven types of biases in RS and providing their definitions and characteristics. Wherein, we specifically provide causality-based explanations for data biases to help the readers to better understand their nature.
- Conducting a comprehensive review and providing a taxonomy of existing methods on recommendation debiasing, as well as discussing their strengths and weaknesses.
- Identifying open challenges and discussing future directions to inspire more research on this topic.

Papers Collection. We collect over 180 papers that analyze the bias issues in recommendation or propose new debiasing methods. We first search the related top-tier conferences and journals to find related work, including WWW, WSDM, SIGIR, KDD, RecSys, CIKM, TOIS, TKDE, etc., with the keywords "recommend", "collaborative filtering", "ranking" or "search" combined with "bias", "fairness" or "exposure" from the year 2010 to 2021. We then traverse the citation graph of the identified papers, retaining the papers that focus bias in RS. Figure 1 illustrates the statistics of collected papers with the publication time and venue.

Survey Audience and Organization. This survey is beneficial for the following researchers and practitioners in RS: 1) who are new to the bias issues and look for a handbook to fast step into this area, 2) who are confused by different bias definitions in the literature and need a systematic study to understand the biases, 3) who want to keep up with the state-of-the-art debiasing technologies in RS, and 4) who face bias issues in building recommender systems and look for suitable solutions. The rest of the survey is organized as follows: Section 2 introduces the preliminaries of RS and the critical issue of feedback loop. Section 3 and 4 are the main content, which summarizes the seven types of biases and provides a taxonomy of debiasing technologies in RS. Section 5 discusses open challenges and future directions, and Section 6 concludes the survey.

2 PRELIMINARIES: RECOMMENDER SYSTEM AND FEEDBACK LOOP

2.1 Feedback Loop in Recommendation

From a bird's-eye view, we can abstract the lifecycle of recommendation as a feedback loop among three key components: User, Data, and Model. As Figure 2 shows, the feedback loop consists of three stages:

- **User→Data** (Collection), which indicates the phase of collecting data from users, including user-item interactions and other side information (*e.g.,* user profile, item attributes, and contexts).
- **Data→Model** (Learning), which represents the learning of recommendation models based on the collected data. At its core is to derive user preference from historical interactions, and predict how likely a user would adopt a target item. Extensive studies have been conducted over past decades.
- **Model→User** (Serving), which returns the recommendation results to users, so as to satisfy the information need of users. This stage will affect the future behaviors and decisions of users.

Through this loop, users and the RS are in a process of mutual dynamic evolution, where personal interests and behaviors of users get updated via recommendation, and the RS can lead to a self-reinforcing pattern by leveraging the updated data.

2.2 Recommendation Task Formulation

We use uppercase character (*e.g.,* U) to denote a random variable; lowercase character (*e.g.,* u) to denote its specific value; and the character in calligraphic font (*e.g.,* $\mathcal{U}$) to represent the space of the corresponding random variable. The probability distribution of a random variable is notated with $p(.)$, while the expectation of a function of a random variable X is notated with $\mathbb{E}_X[.]$.

Suppose we have a recommender system with a user set $\mathcal{U}$ and an item set $\mathcal{I}$. We denote the collected interactions D_T between the user set $\mathcal{U}$ and item set $\mathcal{I}$, as a list of user-item-label triplets (u, i, r), which are drawn from an unknown training distribution $p_T(U, I, R)$. Here $r \in \mathcal{R}$ denotes the feedback label given by a user to an item. It can be explicit (*e.g.,* rating values) that directly reflects user preference on the rated item, or be implicit (*e.g.,* purchase, click, view) indicating whether the user u is willing to interact with the item i. For better description, we use the notation r_{ui} with the subscript denoting the ground-truth label of the user-item pair (u, i), while use the r_{ui}^o denoting the observed label of (u, i) in training data. Besides, we define a Bernoulli random variable $S \in \{0, 1\}$ indicating whether the instance is observed in D_T and use notation s_{ui} for a certain user-item pair (*i.e.,* $s_{ui} = 1$ *iff.* $(u, i, r_{ui}^o) \in D_T$). The task of a recommendation system can be stated as follows: learning a recommendation model from the available dataset D_T so that it can capture user preference and make a high-quality recommendation in the serving stage. Formally, let $\delta(., .)$ denote the error function measuring the distance between the prediction and the ground truth label [3]. The goal of recommendation is to learn a parametric model $f : \mathcal{U} \times \mathcal{I} \to \mathcal{R}$ from D_T to minimize the following *True Risk*:

$$L(f) = \mathbb{E}_{(u,i,r) \sim p_E}[\delta(f(u, i), r)] \tag{1}$$

where $P_E(U, I, R)$ denotes the ideal unbiased data distribution for model testing. This distribution can be factorized as the product of the user-item pair distribution $p_E(U, I)$ (often supposed as uniform) and the factual preference distribution for each user-item pair $p_E(R|U, I)$. As such, the *True Risk* is often written as a metric over all user-item pairs:

$$L(f) = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \mathbb{E}_{r_{ui} \sim p_E(R|U,I)}[\delta(f(u, i), r_{ui})] \tag{2}$$

Since the ideal test distribution is not accessible, the learning is conducted on the training set D_T by optimizing the following *empirical risk*:

$$\hat{L}_T(f) = \frac{1}{|D_T|} \sum_{(u,i):s_{ui}=1} \delta(f(u, i), r_{ui}^o) \tag{3}$$

---

[3] Without loss of generalization, here we just present the point-wise loss. In fact, other types of losses can be extended straightforwardly if we regard the item pair or list as an instance.
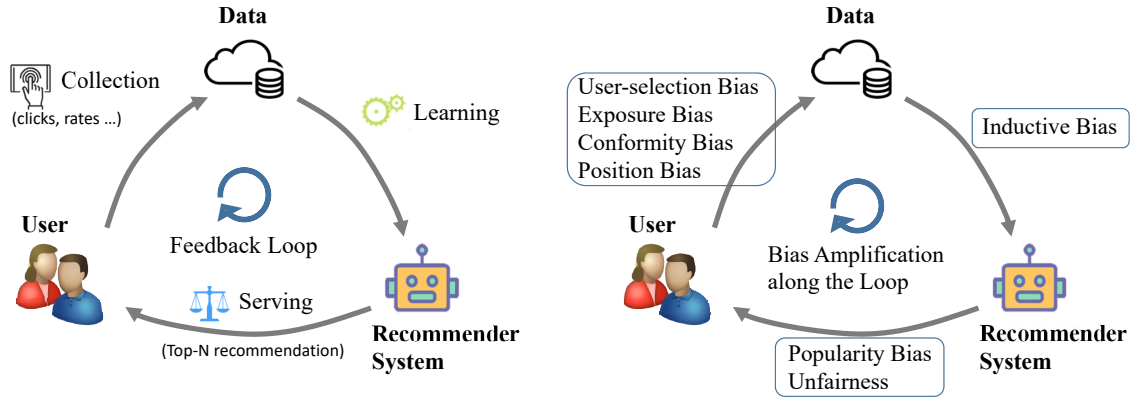
Fig. 2. Feedback loop in recommendation, where biases occur in different stages.

Table 1. The characteristics of seven types of biases in recommendation and the bias amplification in loop.

Types	Stages in Loop	Cause	Effect	Major solutions
Selection Bias	User→Data	Users' self-selection	Skewed observed rating distribution	Data Imputation; Propensity Score; Joint Generative Model; Doubly Robust Model
Exposure Bias	User→Data	Item Popularity; Intervened by systems; User behavior and background	Unobserved interactions do not mean negative	Giving confidence weights by heuristic, sampling or exposure-based model; Propensity Score; Causality-based Model
Conformity Bias	User→Data	Conformity	Skewed interaction labels	Modeling social or popularity effect
Position Bias	User→Data	Trust top of lists; Exposed to top of lists	Unreliable positive data	Click models; Propensity Score; Trust-aware Model
Inductive Bias	Data→Model	Added by researchers or engineers	Better generalization, lower variance or Faster recommendation	-
Popularity Bias	Model→User	Algorithm and unbalanced data	Matthew effect	Regularization; Adversarial Learning; Causal Graph
Unfairness	Model→User	Algorithm and unbalanced data	Unfairness for some groups	Rebalancing; Regularization; Adversarial Learning; Causal Modeling
Bias amplification in Loop	All	Feedback loop	Enhance and spread bias	Break the loop by collecting random data or using reinforcement learning

If the training data and test data are identically and independently distributed (*aka. i.i.d.* assumptions), the empirical risk $\hat{L}_T(f)$ would be an unbiased estimator of the true risk $L(f)$, *i.e.,* $\mathbb{E}_{p_T}[L_T(f)] = L(f)$. The PAC learning theory [62] states that the learned model will be approximately optimal if we have sufficiently large training data set.

3 BIAS IN RECOMMENDATION

In this section, we first summarize and position different types of biases in the feedback loop, as illustrated in Figure 2 and Table 1. We then present in-depth analyses of their relations and discuss how they are intensified along the feedback loop.

3.1 Bias in Data

As the data, of user interactions, are observational rather than experimental, biases are easily introduced into the data. They typically stem from different subgroups of data, and make the recommendation models capture these biases and even scale them, thereby leading to systemic racism and suboptimal decisions. In this subsection,
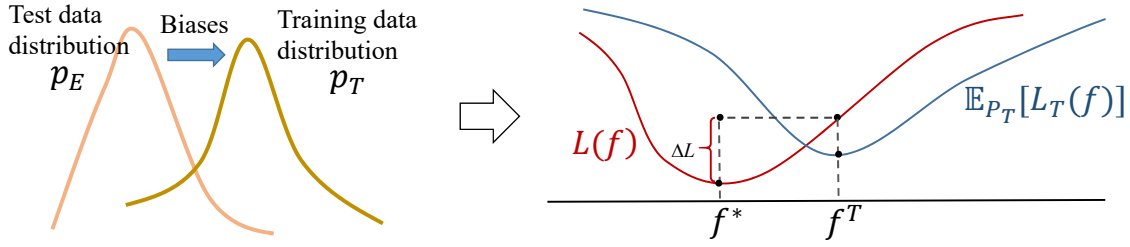
Fig. 3. illustration of the data bias and its negative effect on model training.

we would first give the general definition of the data bias and then categorize it into four groups: selection bias, conformity bias, exposure bias and position bias.

3.1.1 Definition of data bias. The *i.i.d.* assumption lays a foundation for recent learning-based methods to generalize well on the test environment. However, this assumption may not hold in real recommender systems. Typically, the data collection process in RS is observational rather than experimental. The sample selection or user decision would inevitably be affected by many undesirable factors, such as the exposure mechanism of RS or public opinions, making the training data distribution deviate from test distribution. Training data only gives a skewed snapshot of user preference, making the recommendation model sink into sub-optimal result. We name such notorious distribution deviation phenomenon as data bias:

- **Data Bias.** *The distribution for which the training data is collected is different from the ideal test data distribution.*

Figure 3 illustrates the data bias and its negative effect. Bias distorts training distribution, causing the model flow towards wrong direction. The red curve denotes the true risk function for testing, while the blue curve denotes the expected empirical risk function for training. As the two risks are expected over different distribution, they will behave rather differently even in their optimum (*i.e.,* f^* versus f^T). It means that even if a sufficiently large training set is provided and the model arrives at empirical optimal point f^T, there still exists a certain gap ΔL between the optimum $L(f^*)$ and the empirical one $L(f^T)$. Blindly fitting a recommendation model without considering the inherent data bias will result in inferior performance.

In the following, we will introduce four types of data biases, with providing their definitions and characteristics. We also provide causality-based explanations for each bias to help the readers to better understand its nature.

3.1.2 Selection Bias. Selection bias originates from users' numerical ratings on items (*i.e.,* explicit feedback), which is defined as:

- **Selection Bias.** *Selection Bias happens as users are free to choose which items to rate, so that the observed ratings are not a representative sample of all ratings. In other words, the rating data is often missing not at random (MNAR).*

Prior study conducted by Marlin *et al.* [116] offers compelling evidence to show the existence of selection bias in the rating data. In particular, they conducted a user survey to gather the user ratings to some randomly-selected items, as a comparison with that to conventional user-selected items. Figure 4 summarizes the comparison and offers two findings: 1) users tend to select and rate the items that they like; and 2) users are more likely to rate particularly bad or good items. These results suggest that selection bias is inherent in the observed data, since they are missing not at random. The distribution of observed rating data is different from the distribution of all ratings [66, 150].
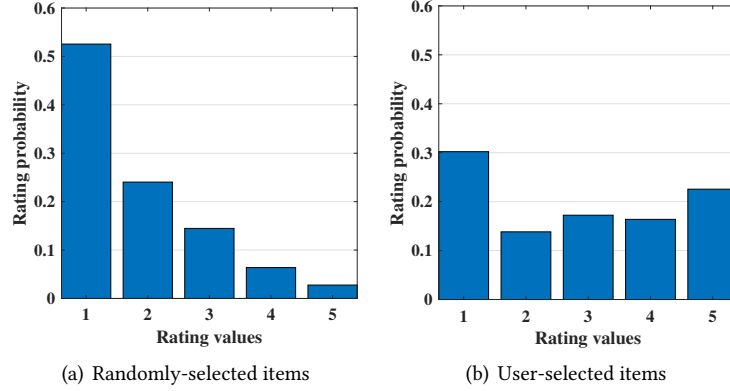
(a) Randomly-selected items

(b) User-selected items

Fig. 4. Distribution of rating values for randomly selected items and user-selected items. The data is from [116] with permission.

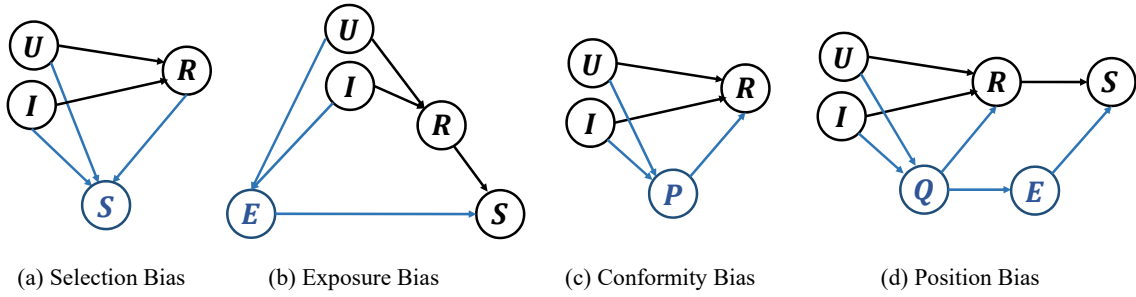(a) Selection Bias   (b) Exposure Bias   (c) Conformity Bias   (d) Position Bias

Fig. 5. Causal graphs of four types of data bias. U: user; I: item; R: feedback label; S: observation variable (S=1, observed; S=0, unobserved); E: exposure; P: public opinion; Q: item position.

Causality-based Explanation. Figure 5(a) illustrates the generative process of the observed rating data. The links $(U, V) \to R$ represent the causal effect of the features of the user U and item I on their interaction label R, for which the recommendation model aims to estimate. In observational study, the collected rating data is not evenly presented and the variables U, I, R would affected the observation of the instance ($S = 1$, observed; $S = 0$, unobserved). This mechanism makes the distribution of the observed rating data (*i.e., $p_T \equiv p(U, I, R|S = 1)$*) inconsistent with the ideal test distribution (*i.e., $p_E \equiv p(U, I, R)$*). The model that directly learned on observed data would suffer.

From another perspective, the causal graph shows two sources of association between the causes (U, I) and the outcome R: 1) the desirable causal effect $(U, V) \to R$; 2) the collision path [132] $(U, I) \to S \leftarrow R$ that links (U, I) and R through their common (conditioned on) effects $S = 1$. The analyses conditioned on $S = 1$ would create spurious association between (U, I) and R. The model learned on observed data would capture skewed patterns.

3.1.3 Exposure Bias. Implicit feedback is widely used in recommendation, which reflects natural behaviors of users, such as purchases, views, clicks. Distinct from explicit feedback that offers numerical ratings, implicit feedback only provides partial signal of positive. As the knowledge about what the user dislikes is not available,

the learning must rely on unobserved interactions, mining the negative signal from them. Exposure bias happens in such one-class data, which is defined as:

- **Exposure Bias.** *Exposure bias happens as users are only exposed to a part of specific items so that unobserved interactions do not always represent negative preference.*

In particular, an unobserved interaction between a user and an item can be attributed to two possible reasons: 1) the item does not match user interest; and 2) the user is unaware of the item. Hence, ambiguity arises in the interpretation of unobserved interactions. The inability to distinguish real negative interactions (e.g. exposed but uninterested) from the potentially-positive ones (e.g. unexposed) will result in severe biases. Previous studies have investigated several dimensions of data exposure: 1) Exposure is affected by the policy of the previous recommender systems, which controls what items to show [107]. Hence, some recent works [107] also name such "exposure bias" as "previous model bias". 2) As users may actively search and find the items of interest, the selection of users is a factor of exposure [125, 169], and makes highly relevant items more likely to be exposed. Hence, in this scenario, "exposure bias" is named as "user-selection bias". 3) The background of users is another factor to expose items, such as social friends [32], communities that they belong to [31], and geo locations. 4) Popular items are more likely to be seen by users. Hence, such "popularity bias" is another form of "exposure bias"[217]. In order to facilitate readers and prevent concept confusion, we use the unified standard definition, "exposure bias", throughout this paper, rather than the separated definitions of the aforementioned factors.

Causality-based Explanation. Figure 5(b) illustrates the generative process of the collected implicit feedback data: the links $(U, I) \rightarrow R$ represent the causal effect of user/item features on the feedback, for which the recommender aims to estimate; the link $R \rightarrow S$ represents the missing mechanism in implicit feedback — only positive interactions are observed $p(R = 1|S = 1) = 1$. As the negative instances are not available, the learning must resort to the unobserved interactions. Recent work on implicit feedback would leverage the observation variable S as a surrogate label — *i.e.,* marking observed interactions as positive while unobserved as negative. It is rational in ideal scenario as the distribution $p(R|U, I)$ is equal to $P(S|U, I)$. However, due to the wide existence of the exposure bias in practical, the equation does not hold. It can be understood from the paths $(U, I) \rightarrow E \rightarrow S$, where the user/item features would affect whether the item is exposed to the user (marked as a variable E); and E would further distort the distribution $P(S|U, I)$, as a user could only generate interactions on the exposed items *i.e.,* $P(S = 0|E = 0) = 1$.

From another perspective, Figure 5(b) shows two sources of associations between the variables (U, I) and S. 1) the desirable causal effect along the $(U, V) \rightarrow R \rightarrow S$; 2) the spurious association created by the exposure bias through the paths $(U, V) \rightarrow E \rightarrow S$. The correlations captured by the recommendation model may fail to reflect the true preference.

3.1.4 Conformity Bias. Different from the aforementioned biases that contribute on the data observation, conformity bias distorts user judgment, which is defined as follow:

- **Conformity Bias.** *Conformity bias happens as users tend to behave similarly to the others in a group, even if doing so goes against their own judgment, making the feedback do not always signify user true preference.*

For example, influenced by high ratings of public comments on an item, one user is highly likely to change her low rate, avoiding being too harsh [110, 166]. Such phenomenon of conformity is common and cause biases in user ratings [95]. As shown in Krishnan *et al.* [92], user ratings follow different distributions when users rate items before or after being exposed to the public opinions. Moreover, conformity bias might be caused by social influence, where users tend to behave similarly with their friends[24, 112, 155, 172]. Hence, the observed interactions are skewed and might not reflect users' real preference on items [110].

Causality-based Explanation. As illustrated in Figure 5(c), conformity bias can be understood from the additional causal path $I \rightarrow P \rightarrow R$. Public opinion P (*e.g.,* popularity, averaged rating) depends on item property (I)

and impacts user judgment R. This undesirable phenomenon would distort the conditional distribution $p(R|U, I)$, making the training distribution deviate from reflecting user true preference.

From another perspective, conformity bias creates spurious association between (U, I) and R through the path $I \rightarrow P \rightarrow R$. The model trained on observed data would easily capture spurious association, leading to poor performance.

3.1.5 Position Bias. Position bias is very common in recommendation, particularly in advertisement system or search engine:

- **Position Bias.** *Position bias happens as users tend to interact with items in higher position of the recommendation list regardless of the items' actual relevance so that the interacted items might not be highly relevant.*

Here "relevance" is widely used in the field of information retrieval, which denotes how the items are preferred by the users. Popularity bias happens in implicit feedback data and describes a tendency of users to notice or interact with items in certain positions of lists with higher probability, regardless of the items' actual relevance[40]. For example, recent studies on eye tracking demonstrate that users are less likely to browse items that are ranked lower in vertical lists, while they only examine the first few items at the top of lists [75, 76]. Moreover, Maeve *et al.* [127] shows that users often trust the first few results in the lists and then stop assessing the rest, without evaluating the entire list holistically [89]. As such, the data collected from user feedback towards the recommended lists may fail to reflect user preference faithfully [40].

Causality-based Explanation. Figure 5(d) illustrates the generative process of users' feedback on recommendation lists, where each item is companied with a position Q displayed in the previous recommendation list. The effect of position Q on outcome S is multi-folded: 1) along the path $Q \rightarrow E \rightarrow S$, the display position would impact the probability that the item is exposed to the user; 2) along the path $Q \rightarrow R \rightarrow S$, the position would also hinder users' own judgement, as users trust the recommender system and may over-estimate the relevance of the highly-ranked items. As such, position bias is quite complex and would skew both the data observation $p(S|U, I, R)$ as well as the user judgment $p(R|U, I)$, making the training distribution deviate significantly from the ideal test one.

From another perspective, position bias creates two spurious associations between (U, I) and S through the paths $(U, I) \rightarrow Q \rightarrow E \rightarrow S$ and $(U, I) \rightarrow Q \rightarrow R \rightarrow S$, which should be conquered.

3.2 Bias in Model

Bias is not always harmful. In fact, a number of inductive biases have been added deliberately into the model design to achieve some desirable characteristics:

- **Inductive Bias.** *Inductive bias denotes the assumptions made by the model to better learn the target function and to generalize beyond training data.*

The ability to generalize the prediction to unseen examples is the core of machine learning. Without assumptions on the data or model, generalization cannot be achieved since the unseen examples may have an arbitrary output space. Similarly, building a RS needs to add some assumptions on the nature of the target function. For example, Johnson *et al.*[78] assumes an interaction can be estimated by embedding inner product, while He *et al.* [64] adopts the neural network as its better generalization. Besides target function, inductive bias have been added in other aspects. An example is the adaptive negative sampler[47, 130, 137, 162], which aims to over-sample the "difficult" instances in order to increase learning speed, even though the resultant loss function will differ significantly from the original. Another example is the discrete ranking model[101, 201, 218] which embeds user and items as binary codes to improve the efficiency of recommendation, which is at the expense of sacrificing the representation ability.
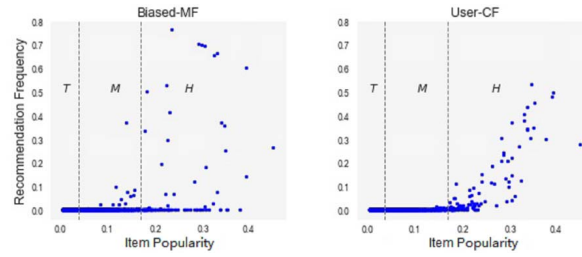
Fig. 6. Item popularity VS. recommendation frequency (Biased-MF [90] and User-CF [9]), where items were classified into three different groups: H denoting the set of most popular items that take up around 20% entire ratings, T denoting the set of most unpopular items that take up 20% entire ratings and M denotes the rest. The figure was reproduced from [3] with authors' permission.

3.3 Bias and Unfairness in Results

Besides the aforementioned biases in data or model, two important biases in recommendation results have been studied, which are defined as follows:

- **Popularity Bias.** *Popular items are recommended even more frequently than their popularity would warrant [3].*

The long-tail phenomenon is common in RS data: in most cases, a small fraction of popular items account for the most of user interactions [3]. When trained on such long-tailed data, the model usually gives higher scores to popular items than their ideal values while simply predicts unpopular items as negative. As a result, popular items are recommended even more frequently than their original popularity exhibited in the dataset. Popularity bias has been empirically verified by Abdollahpouri *et al.* [3, 4]. Figure 6 shows relationship between item popularity and recommendation frequency. We can find most of recommended items are located at high popularity area (H). In fact, they are recommended to a much greater degree than even what their initial popularity warrants [3].

Ignoring the popularity bias results in many issues: 1) It decreases the level of personalization and hurts the serendipity. Since the preferences of different users are diverse, always recommending popular items will hurt user experience, especially for the users favoring niche items. 2) It decreases the fairness of the recommendation results [5]. Popular items are not always of high quality. Over-recommending popular items will reduce the visibility of other items even if they are good matches, which is unfair. 3) Popular bias will further increase the exposure opportunities of popular items, making popular items even more popular – the collected data for future training becomes more unbalanced, raising the so-called "Matthew effect" issue [220].

Another type of bias arises in the recommendation results is unfairness. Fairness has attracted increasing attention in recent years. A consensual definition of fairness is "*absence of any prejudice or favoritism towards an individual or a group based on their intrinsic or acquired traits*" [119], and the unfairness can be defined as follow:

- **Unfairness.** *The system systematically and unfairly discriminates against certain individuals or groups of individuals in favor others [54].*

Unfairness issue has been an obstacle to making recommender systems more entrenched within our society. In particular, based on attributes like race, gender, age, education level, or wealth, different user groups are usually unequally represented in data. When training on such unbalanced data, the models are highly likely to learn these over-represented groups, reinforce them in the ranked results, and potentially result in systematic discrimination and reduced the visibility for disadvantaged groups (*e.g.,* under-representing the minorities, racial or gender stereotypes) [104]. For example, in the context of job recommendation, previous work [43, 94] found that, compared to men, women saw less ads about high paying jobs and career coaching services, which is
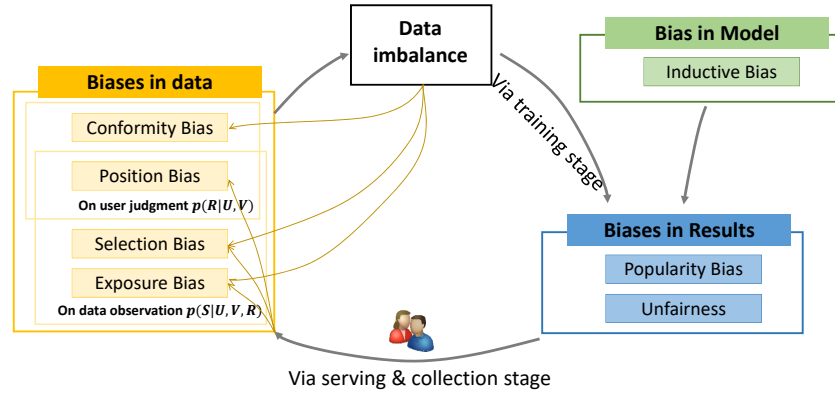
Fig. 7. Relations between seven types of biases.

caused by gender imbalance. Analogously, friend recommendation in social graphs may reinforce historical biases towards a majority and prevent minorities from being social influencers with high reach [85, 151]. Another similar issue has been found in book recommendation, where the methods prefer recommending books of male authors [51]. Analyzing the unfairness issue inherent in recommendation is therefore becoming essential and desirable.

3.4 Feedback Loop Amplifies Biases

Real-world recommender systems usually create a pernicious feedback loop. Previous subsections summarize the biases occurred in different stages of the loop, while these biases could be further intensified over time along the loop. To better understand this effect, figure 7 illustrates the relations of seven types of biases. Data biases would incur or intensify the data imbalance, exacerbating bias issues in recommendation results (*Data bias → Data imbalance → Bias in Results*); while the biased recommendations would in turn impact the decisions, exposure, and selections of users, reinforcing the biases in users' future behaviors (*Bias in Results → Data Bias*). Taking the position bias as an example, top items typically benefit from a greater volume of traffic, which in turn increases their ranking prominence and the volume of traffic they receive, resulting in a rich-get-richer scenario[127]. Many researchers also study the impact of feedback loop on the popularity bias [25, 71, 114]. Their simulated results show that feedback loop will amplify popularity bias, where popular items become even more popular and non-popular items become even less popular. These amplified biases also will decrease the diversity and intensify the homogenization of users, raising the so-called "echo chambers" or "filter bubbles" [56, 73].

Interestingly, figure 7 also shows that some data biases can be self-reinforced (*Data bias → Data imbalance →Data bias*). Taking conformity bias as an example, as users tend to behave similarly to the major groups, the extent of the data imbalance would increase with time going by, which in turn impacts and exacerbates the conformity bias. Similar situations apply to the selection bias and exposure bias which are also affected by the data imbalance.

4 DEBIASING METHODS

A large number of methods have been proposed to mitigate the effects of bias or unfairness. Table 2 lists the reviewed methods, as well as their strengths and weaknesses. we classify them according to which biases they addressed and which types of methods they adopted.

Table 2. A lookup table for the reviewed methods for recommendation debiasing.

Addressed issues	Categories		How?	Strengths?	Weaknesses?	Publications
Selection Bias	Evaluator	Propensity Score	Weight the data	General, Theoretical-soundness	Requiring proper propensities	[143]
		ATOP	Specific design	Theoretical-soundness	Requiring two strong assumptions	[149]
	Training	Joint Generative Model	Model missing mechanism	Explainable	Requiring assumptions on data generation, Hard to train	[30, 66, 87, 115, 116, 178] [189]
		Data Imputation	Impute pseudo-labels	Simple	Highly sensitive to pesduo-labels	[140, 149, 150]
		Propensity Score	Weight the data	Theoretical-soundness	High variance, Requiring proper propensities	[143, 177]
		Doubly Robust Model	Impute+Weight	Theoretical-soundness, Robust	Requiring proper propensities or pesduo-labels	[176]
Conformity Bias	Modeling popularity influence		Disentangle conformity effect from user preference	Explainable	Requiring assumptions on data generation	[110, 215, 217]
	Modeling social influence		Disentangle social effect from user preference	Explainable	Requiring assumptions on data generation	[24, 112, 155, 172]
Exposure Bias	Evaluator	Propensity Score	Weight the data	Theoretical-soundness	Requiring proper propensities	[190]
	Training	Heuristic	Down-weight unobserved data heurstically	Simple	Coarse-grained, Heuristical	[45, 65, 68, 128, 129, 194] [100, 141]
		Sampling	Down-weight unobserved data via sampling	Efficient	Coarse-grained, Requiring heuristic or side information	[32, 46, 47, 138, 175, 194]
		Exposure-based model	Weight the data via exposure model	Explainable, Learn flexible weights	Hard to train	[29, 31, 32, 102, 163]
		Propensity Scores	Weight the observed data	Theoretical-soundness	Requiring proper propensities, High variance	[141, 219]
		Causality-based Methods	Remove spurious associations via causal inference	Explainable	Requiring assumptions on data generation	[108, 187, 191, 208]
		Others	-	-	-	[113, 125, 126, 169, 204] [12, 42, 142, 181]
Position Bias	Click Models		Model the generative process of clicks	Explainable	Requiring assumptions on data generation, hard to train	[26, 41, 48, 59, 103, 205] [74, 224]
	Propensity Score		Weight the data	Theoretical-soundness	High variance, Requiring proper propensities, Fail to model trust	[6, 67, 77, 136, 144, 153, 169] [10, 52, 77, 77, 134, 157, 171] [8, 35, 60]
	Trust-aware Models		Introduce offset terms to remove trust effect	Theoretical-soundness, Capture trust effect	High variance, Requiring proper propensities, Fail to model trust	[7, 158]
For multiple data biases and their combinations	Universal model		Transfer the knowledge from unbiased data to perform debiasing	Universal, Adaptive	Requiring a set of unbiased data	[17, 28, 106, 107]
Popularity Bias	Regularization		Introduce regularization terms	Simple, Straightforward	Possibly hurt accuracy	[2, 38, 83, 179, 221]
	Adversarial Learning		Leverage adversary to bridge the gap between niche and popular items	Balancing representation	Possibly hurt accuracy	[91]
	Causal Graph		Leverage causal graph to elucidate and mitigate popularity bias	Explainable	Requiring assumptions on data generation	[167, 180, 208, 215, 217]
	Others		-	-	-	[1, 19]
Unfairness	Rebalancing		Directly balance the data or recommendation results	Straightforward	Possibly hurt accuracy	[11, 15, 57, 57, 98, 133, 199] [22, 53, 135, 146]
	Regularization		Formulate the fairness criteria as a regularizer	Straightforward	Possibly hurt accuracy	[80–82, 84, 147, 188, 200] [2, 14, 21, 105, 192, 193]
	Adversarial Learning		Leverage adversary to isolate the effect of sensitive attributes	Fair representation	Possibly hurt accuracy	[13, 18, 50, 99, 183]
	Causal Modeling		Estimate fairness with intervening sensitive attributes	Explainable, Counterfactual fairness	Requiring assumptions on data generation	[93, 93, 122, 185, 186, 202]
	Others		-	-	-	[55, 69, 97]
Loop effect	Uniform Data		Intervene in the system with a random logging policy	Straightforward, Effective	Hurting the user experience and the system profit	[17, 73, 107, 107, 139, 196] [28, 195]
	Reinforcement learning		Intervene in the system with a smarter strategy for long-term benefits	Adaptively balancing exploration-exploitation	Hard to train, Off-policy evaluation is chanllenging	[36, 96, 161, 173, 209, 213, 214] [27, 34, 70, 170, 210–212, 216] [70, 72, 117, 154]
	Others		-	-	-	[148, 152]

4.1 Methods for Selection Bias

Training and testing a recommendation model on the observed rating data will suffer from the selection bias, as the observed ratings are not a representative sample of all ratings. Here we fist introduce how to evaluate a recommendation model under biased rating data, and then review four kinds of methods that mitigates selection bias on recommender training.

4.1.1 Debiasing in evaluation. Given a recommendation model, we want to evaluate its performance on rating prediction or recommendation accuracy. Standard evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Discounted Cumulative Gain@k (DCG@k) or Precision (Pre@k) can be written as[143]:

$$H(\hat{R}) = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \delta(\hat{r}_{ui}, r_{ui}) \tag{4}$$

for an appropriately chosen $\delta(r_{ui}, \hat{r}_{ui})$:

$$\text{MAE:} \ \delta(r_{ui}, \hat{r}_{ui}) = |r_{ui} - \hat{r}_{ui}| \tag{5}$$

$$\text{MSE:} \ \delta(r_{ui}, \hat{r}_{ui}) = (r_{ui} - \hat{r}_{ui})^2 \tag{6}$$

$$\text{DCG@k} : \delta(r_{ui}, \hat{r}_{ui}) = (1/\log(\text{rank}(\hat{r}_{ui}))) \, r_{ui} \tag{7}$$

$$\text{Pre@k} : \delta(r_{ui}, \hat{r}_{ui}) = (1/k)r_{ui} \cdot \mathbf{I}[\text{rank}(\hat{r}_{ui}) \le k] \tag{8}$$

where $\mathbf{I}[.]$ denotes indicator function ($\mathbf{I}[.] = 1$ iff the internal condition holds), r_{ui} denotes the true rating values of the item i given by the user u and $\hat{r}_{ui}$ denotes the predicted rating values by the recommendation model. As true ratings r are usually partially observed [4]. The conventional evaluation usually use the average over only the observed entries:

$$\hat{H}_{\text{naive}}(\hat{r}) = \frac{1}{|\{(u,i) : s_{ui} = 1\}|} \sum_{(u,i):s_{ui}=1} \delta(\hat{r}_{ui}, r_{ui}) \tag{9}$$

where s_{ui} denotes the number of observed ratings in the dataset. We can find $\hat{H}_{\text{naive}}(\hat{r})$ is not an unbiased estimate of the true performance [143]:

$$E_O\left[\hat{H}_{\text{naive}}(\hat{r})\right] \ne H(\hat{r}) \tag{10}$$

where $\hat{H}_{\text{naive}}(\hat{r})$ is expected over the observation probability. The gap is caused by selection bias, making the observed ratings not a representative sample of all ratings. Two strategies have been presented in recent work.

Propensity Score. To remedy the selection bias in evaluation, some recent work [143] considers a recommendation as an intervention analogous to treating a patient with a specific drug. In both tasks, we have only partial knowledge of how much certain patients (users) benefit from certain treatments (items), while the outcomes for most patient-treatment (user-item) pairs are unobserved. A promising strategy for both tasks is weighting the observations with inverse propensity scores. The propensity ρ_{ui}, which is defined as the marginal probability of observing a rating value ($\rho_{ui} = p(s_{ui} = 1)$) for certain user-item pair (u, i), can offset the selection bias. The proposed estimator is defined as:

$$\hat{H}_{IPS}(\hat{r} \mid \rho) = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{(u,i):s_{ui}=1} \frac{\delta(\hat{r}_{ui}, r_{ui})}{\rho_{ui}} \tag{11}$$

---

[4]Recent work on selection bias usually assumes the conditional distribution $p(R|U, I)$ is stable (*i.e.,* $P_E(R|U, I) = P_T(R|U, I)$). Therefore, observed rating values can be considered as true ones $r_{ui} = r_{ui}^o$.

which is an unbiased estimator of the ideal metric:

$$
\begin{aligned}
\mathbb{E}_S\left[\hat{H}_{IPS}(\hat{r}\mid\rho)\right] &= \frac{1}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\mathbb{E}_S\left[\frac{\delta(\hat{r}_{ui},r_{ui})}{\rho_{ui}}s_{ui}\right]\\
&= \frac{1}{|\mathcal{U}||\mathcal{I}|}\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}}\delta(\hat{r}_{ui},r_{ui}) = H(\hat{r})
\end{aligned}
\tag{12}
$$

ATOP. Steck *et al.* [149] propose another unbiased metric ATOP to evaluate recommendation performance with two mild assumptions: (1) the relevant (high) rating values are missing at random in the observed data; (2) Concerning other rating values, we allow for an arbitrary missing data mechanism, as long as they are missing with a higher probability than the relevant rating values. They define the ATOP as:

$$
\text{TOPK}_u^{\text{obs}}(k) = \frac{N_u^{+,\text{obs},k}}{N_u^{+,\text{obs}}}
\tag{13}
$$

$$
\text{TOPK}^{\text{obs}}(k) = \sum_u w^u \text{TOPK}_u^{\text{obs}}(k)
\tag{14}
$$

$$
\text{ATOP}^{\text{obs}} = \int_0^1 \text{TOPK}^{\text{obs}}(k)dk
\tag{15}
$$

which computed from biased explicit feedback data and $N_u^{+,\text{obs}}$ denotes the number of observed relevant (preferred) items of the user u and $N_u^{+,\text{obs},k}$ counts the relevant ones in the top k. The authors prove $\text{ATOP}_u^{\text{obs}}$ is an unbiased estimate of the average recall and proportional to the precision averaged over users.

Discussion. Propensity scores and ATOP are two subtle strategies to remedy selection bias, but they still have two serve weaknesses. The unbiasedness of the IPS-based estimator is guaranteed only when the true propensities are available[140]. The IPS estimator will still be biased if the propensities are specified unproperly. The unbiasedness of the ATOP is guaranteed only when the two assumptions hold. In practice, the missing mechanism is often complex and the assumptions are not always valid. Developing a robust and effective remains a challenge.

4.1.2 Debiasing in model training. In the following, we will review four kinds of methods on mitigating selection bias on recommender training.

Joint Generative Model. Note that the main reason for the selection bias is that users are free to deliberately choose which items to rate. Thus, a straightforward strategy for mitigating selection bias is to jointly consider both rating prediction task ('which rating value the user gives', *i.e.*, r_{ui}) and missing data prediction task ('which items the user select to rate', *i.e.*, s_{ui}). Some recent work [30, 66, 87, 115, 116] propose to jointly model the generative process of rating values and the missing mechanism. Their generative process can be depicted with Figure 5(a), with the assumption that the probability of users' selection on items (*i.e.*, s_{ui}) depends on users' rating values for that item r_{ui}. Correspondingly, s_{ui} has been modeled dependent on r_{ui} with a mixture of Multinomials[116], Logit model[115, 189], MF model [66, 178], binomial mixture model [87], or social-enhanced model [30]. In this way, user's preference can not only learn from rating values but also from the missing mechanism.

Although this kinds of methods is explainable and sometimes effective in some scenarios, jointly modeling the missing mechanism and rating values will lead to a highly complex model, which is hard to be trained. What's worse, the architecture of missing data models are usually heuristically designed. The hypothesis on distribution may not hold in some real cases.

Data Imputation. Note that the inherent nature of selection bias is that the data is missing not random. A straightforward solution for selection bias is to impute the missing entries with pseudo-labels, such that the observed data distribution $p(U, I | S = 1)$ is close to the ideal uniform one $P(U, I)$. For example, Steck *et*

al. [149, 150] propose a light imputation strategy that directly impute the missing data with a specific value r_0, with optimizing the following objective function:

$$\hat{L}_{DI} = \sum_{u \in \mathcal{U}, i \in \mathcal{I}} W_{ui} \cdot \left(r_{ui}^{o\&i} - \hat{r}_{ui}\right)^2 \tag{16}$$

where $r_{u,i}^{o\&i}$ denotes observed or imputed ratings, while $\hat{r}_{u,i}$ denotes the predicted ratings. W_{ui} is introduced to downweight the contribution of the missing ratings.

However, as imputed rating values are specified in a heuristic manner, this kind of methods will suffer from empirical inaccuracy due to inaccurate imputed rating values. Such inaccuracy will be propagated into recommendation model training, resulting in sub-optimal recommendation performance[176].

To resolve this issue, Saito *et al.* [140] propose to learn imputation values with an asymmetric tri-training framework. They first pre-train two predictors (A1,A2) with two specific recommendation models to generate a reliable dataset with pseudo-ratings and then trained a target recommendation model A0 on the pseudo-ratings. Theoretical analysis presented in [140] shows that the proposed method optimizes the upper bound of the ideal loss function. However, the performance of asymmetric tri-training depends on the quality of pre-trained predictor A2, while a satisfied A2 itself is hard obtained from biased data. Nevertheless, model-based imputation strategy is a promising direction for mitigating selection bias, which deserve future exploring.

Propensity Score. Besides on evaluation, propensity score can be utilized to mitigate selection bias on model training [143, 177]. This kind of methods directly use the IPS-based unbiased estimator as the objective and optimize the following risk function:

$$\hat{L}_{IPS} = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{(u,i):s_{ui}=1} \frac{\delta\left(\hat{r}_{ui}, r_{ui}^o\right)}{\rho_{ui}} \tag{17}$$

Except for the propensities $\rho_{ui} = p(s_{ui} = 1)$ that act like weights for each loss term, the training objective is identical to the standard recommendation objective. Also, thanks to the propensities, the selection bias can be mitigated as the IPS-based estimator is an unbiased estimation of the True Risk:

$$\mathbb{E}[\hat{L}_{IPS}] = \mathbb{E}_{S,R}\left[\frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \frac{s_{ui}\delta\left(\hat{r}_{ui}, r_{ui}^o\right)}{\rho_{ui}}\right] = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \mathbb{E}_{r_{ui}^o \sim p_E(R|U,I)}\left[\delta\left(\hat{r}_{ui}, r_{ui}^o\right)\right] = L \tag{18}$$

However, as discussed in the previous subsection, specifying appropriate propensity scores is critical. The performance of IPS-based model depends on the accuracy of the propensities. Moreover, propensity-based methods usually suffer from high variance[140], leading to non-optimal results especially when the item popularity or user activeness is highly skewed.

Doubly Robust Model. As data imputation-based models often have a large bias due to mis-specification while IPS-based model usually suffer from high variance, Wang *et al.* [176] propose to combine the two kinds of models and enjoy a desired double robustness property: the capability to remain unbiased if either the imputed errors or propensities are accurate. They define the following objective function:

$$\mathcal{E}_{\text{DR}} = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \left(\delta\left(\hat{r}_{ui}, r_{ui}^i\right) + \frac{s_{ui}(\delta\left(\hat{r}_{ui}, r_{ui}^i\right) - \delta\left(\hat{r}_{ui}, r_{ui}^o\right))}{\rho_{ui}}\right) \tag{19}$$

where r_{ui}^i denotes the imputed value for certain user-item pair (u, i). The theoretical and empirical analyses presented in [176] validate superiority over both IPS-based and imputation-based models.

Although the model is more robust than single method, it still requires relatively accurate propensity score or imputation data, which is usually hard to specify. Otherwise, its performance also suffers.

4.2 Methods for Conformity Bias

Conformity bias occurs as users are normally influenced by others opinion so that the rating values are deviated from users' true preference. Two types of methods have been proposed to address the conformity bias. The first type of work considers users' behaviors conform to public opinions. For example, Liu *et al.* [110] directly leverage three important features c_{ui}, a_{ui}, d_{ui} in the base recommendation model, where c_{ui} is the number of ratings for item i before user u rates it, a_{ui} is the average rating and d_{ui} is the rating distribution. The predicted rating is generated from XGBoost [37]:

$$\hat{r}_{ui} = xgb \left(\left\{ (1 - \omega) \cdot t_{ui} + \omega \cdot a_{ui}, c_{ui}, a_{ui}, d_{ui} \right\}, \Theta_{xgb} \right) \tag{20}$$

where t_{ui} denotes the prediction returned by basic recommendation model and ω controls the strength of conformity. This way, we can disentangle the effect caused by conformity bias from users' true preference and make a recommendation accordingly. Some recent work further study conformity bias in a more fine-grained manner. Zheng *et al.*[217] propose to model personalized conformity effect as users have different sensitivities to the public opinions; while Zhao *et al.* [215] model time-aware conformity effect by considering item dynamic popularity.

The other type of methods treat user's rating values as synthetic results of user preference and social influence [24, 112, 155, 172]. Thus, similar to [110], they directly leverage social factors in the base recommendation model to generate final prediction and introduce specific parameters to control the effect of social conformity bias.

4.3 Methods for Exposure Bias

Exposure bias occurs as users are only exposed to a part of items so that unobserved interactive data does not always mean negative signal. Exposure bias will mislead both the model training and evaluation. Here we review the work on correcting exposure bias.

4.3.1 Debiasing in evaluation . A straightforward strategy for debiasing in RS evaluation is using the inverse propersity score, which also has been applied to address the selection bias. Yang *et al.* [190] first illustrate evaluation bias in terms of conventional metrics such as AUC, DCG@k, Recall@k on the implicit feedback data, and leverage the IPS framework to offset the exposure bias. They abstract the ideal recommendation evaluator as:

$$R(\hat{Z}) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u|} \sum_{i \in \mathcal{G}_u} c\left(\hat{Z}_{ui}\right) \tag{21}$$

where $\hat{Z}_{ui}$ is the predicted ranking of item i for user u returned by the recommendation model and $\mathcal{G}_u$ denotes the set of all relevant items for user u. Function $c(.)$ needs to be adapted for different metrics, such as:

$$\text{AUC} : c\left(\hat{Z}_{ui}\right) = 1 - \frac{\hat{Z}_{u,i}}{|I|} \tag{22}$$

$$\text{DCG} : c\left(\hat{Z}_{ui}\right) = \frac{1}{\log_2\left(\hat{Z}_{ui} + 1\right)} \tag{23}$$

$$\text{DCG@k} : c\left(\hat{Z}_{ui}\right) = \frac{1\left\{\hat{Z}_{ui} \le k\right\}}{\log_2\left(\hat{Z}_{ui} + 1\right)} \tag{24}$$

$$\text{Recall@k} : c\left(\hat{Z}_{ui}\right) = 1\left\{\hat{Z}_{ui} \le k\right\} \tag{25}$$

However, due to the exposure bias, only partial preferred items are observed, making the model often be evaluated on the biased implicit feedback as:

$$\hat{R}_{\text{AOA}}(\hat{Z}) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{G}_u^*|} \sum_{i \in \mathcal{G}_u^*} c\left(\hat{Z}_{ui}\right)$$

$$= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{\sum_{i \in \mathcal{G}_u} s_{ui}} \sum_{i \in \mathcal{G}_u} c\left(\hat{Z}_{ui}\right) \cdot s_{ui}$$

(26)

where $\mathcal{G}_u^*$ denotes the preferred items that have been exposed to the user u. As users usually have biased exposure, the output of AOA evaluator does not conform the true performance, i.e. $\mathbb{E}_O\left[\hat{R}_{\text{AOA}}(\hat{Z})\right] \neq R(\hat{Z})$.

To address this problem, similar to the treatment for selection bias in explicit feedback data, Yang *et al.* [190] propose to weight the each observation with the inverse of its propensity for implicit feedback data. The intuition is to down-weight the commonly observed interactions, while up-weighting the rare ones. Thus, the IPS-based unbiased evaluator is defined as follow:

$$\hat{R}_{\text{IPS}}(\hat{Z} \mid \rho) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{G}_u|} \sum_{i \in G_u^*} \frac{c\left(\hat{Z}_{ui}\right)}{\rho_{ui}}$$

$$= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{G}_u|} \sum_{i \in \mathcal{G}_u} \frac{c\left(\hat{Z}_{ui}\right)}{\rho_{ui}} \cdot s_{ui}$$

(27)

which is unbiased estimator of the ideal metrics, i.e. $E_O\left[\hat{R}_{\text{IPS}}(\hat{Z} \mid \rho)\right] = R(\hat{Z})$.

4.3.2 Debiasing in model training. To deal with the exposure bias and extract negative signal from the implicit feedback, a conventional strategy is treating all the unobserved interactions as negative and specify their confidence. The objective function of most such methods can be summarized as follow:

$$\hat{L}_W = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} W_{ui} \delta\left(\hat{r}_{ui}, s_{ui}\right)$$

(28)

where s_{ui} is a surrogate label indicating whether the interaction between user u and item i is observed or not; W_{ui} denotes the confidence weight, controlling the confidence that the the feedback of user-item pair (u, i) should be predicted as s_{ui}. The specification of the confidence weight is critical to the model performance and can be roughly categorized into three types:

Heuristic Weighting. The first is heuristic-based strategy. For example, the classic weighted factorization matrix (WMF) [68] and dynamic MF [45] used a simple heuristic that the un-observed interactions are assigned with a uniform lower weight, i.e., $W_{ui} = 1$ for $s_{ui} = 1$ and $W_{ui} = c$ $(0 < c < 1)$ for $s_{ui} = 0$. The intuition behind this strategy is that unobserved data is relatively unreliable, which can be attributed to dislike or unknown; Some researchers [128, 129] specify the confidence with based on user activity level, i.e., $W_{ui} = c_u, c_u \propto \sum_i s_{ui}$, as users associate with more items provide more reliable information; Analogously, item popularity has been considered to specify confidence weights [65, 194], as popular items are more probable to be exposed; Also, user-item feature similarity [100] has been considered to define the confidence.

However, assigning appropriate confidence weights heuristically is challenging, as the optimal data confidence may change for different user-item combinations. Choosing confidence weights usually require rich human expertise or large computational resource for grid search. Furthermore, it is unrealistic to manually set flexible and diverse weights for millions of data. Coarse-grained confidence weights will create empirical bias on estimating user's preference.

Sampling. Another solution to address exposure bias is performing sampling. The sampling strategy determines which data are used to update parameters and how often, and thus scale the data contribution. Provided the sampled probability of an instance is p_{ui}, learning a recommendation model with sampling is equivalent to learning the model with the following weighted objective function:

$$\mathbb{E}_{(u,i)\sim p}[\delta\left(\hat{r}_{ui}, s_{ui}\right)] = \sum_{u\in\mathcal{U}, i\in\mathcal{I}} p_{ui}\delta\left(\hat{r}_{ui}, s_{ui}\right) \tag{29}$$

where the sampling distribution acts as data confidence weights. Sampling strategy has been widely applied as its efficiency. For example, Logistical matrix factorization [78], BPR [138], or most of neural-based recommendation models (e.g. CDAE [184], NCF [64], LightGCN [63]) apply the uniform negative sampler; Yu *et al.* [194] considers to over-sample the popular negative items, as they are more likely to be exposed. However, these heuristic samplers are insufficient to capture real negative instances. Thus, some researchers explore to leverage side information to enhance the sampler. Ding *et al.* [46, 47] leverage viewed but non-clicked data to evaluate user's exposure; Chen *et al.* [32] leverage social network information in their sampling distribution; Wang *et al.* [175] construct an item-based knowledge graph and perform sampling on the graph.

Exposure-based model. Another strategy is to develop an exposure-based model, which is capable of capturing how likely a user is exposed to an item[29, 163]. EXMF [102] introduces an exposure variable and assumes the following generative process of implicit feedback:

$$e_{ui} \sim Bernoulli(\eta_{ui}) \tag{30}$$

$$(s_{ui}|e_{ui} = 1) \sim Bernoulli(\hat{r}_{ui}) \tag{31}$$

$$(s_{ui}|e_{ui} = 0) \sim \delta_0 \tag{32}$$

where e_{ui} denotes whether a user u has been exposed to the item i; δ_0 denotes delta function $p(s_{ui} = 0|e_{ui} = 0) = 1$ and can be relaxed as Bernoulli distribution parameterized with a small value; η_{ui} is the prior probability of exposure. When $e_{ui} = 0$, we have $s_{ui} \approx 0$, since when the user does not know the item he can not interact with it. When $e_{ui} = 1$, *i.e.,* the user has known the item, he will decide whether or not to choose the item based on his preference. s_{ui} can be generated with normal recommendation model. In this way, by optimizing the marginal probability, the model can adaptively learn the exposure probability, which will be transformed as confidence weights to remedy exposure bias. Chen *et al.*[31, 32] give detailed analyses of EXMF and rewrite the objective function of EXMF as follows:

$$\hat{L}_{EXMF} = \sum_{u\in\mathcal{U}, i\in\mathcal{I}} \gamma_{ui}\delta(\hat{r}_{ui}, x_{ui}) + \sum_{u\in\mathcal{U}, i\in\mathcal{I}} g(\gamma_{ui}) \tag{33}$$

where γ_{ui} is defined as variational parameters of the user's exposure. $g(\gamma_{ui})$ is a γ_{ui}-dependent function:

$$g(\gamma_{ui}) = (1 - \gamma_{ui})\ell(x_{ui}, \varepsilon) + \ell(\gamma_{ui}, \eta_{ui}) - \ell(\gamma_{ui}, \gamma_{ui}) \tag{34}$$

where $\ell(a, b) = alog(b) + (1 - a)log(1 - b)$. We can find γ_{ui}, which indicates how likely a user is exposed to an item, acts as confidence weights to control the contribution of the data on learning a recommendation model. This finding is consistent with our intuition. Only if the user has been exposed to the item, can he decide whether or not to consume the items based on his preference. Thus, the data with larger exposure are more reliable in deriving user preference.

However, directly estimating data confidence from Equation (33) is insufficient as the model will easily suffer from over-fitting and inefficiency problems due to the large scale of the inferred parameters γ. A promising solution is to re-parameterize the confidence weights with a simpler function. For example, some researchers propose to infer confidence wights with a social-based [32] or community-based model [31, 160].

Propensity Score. Although the aforementioned weighting strategies are popular and have been studied for a long time, Saito *et al.* [141] argue that these methods can not address exposure bias entirely — For any choice of the weights ($\forall W_{ui} \in \mathbb{R}$), The weighted empirical risk L_W can not be an unbiased estimator of the ideal *True Risk, i.e.,* $\mathbb{E}[\hat{L}_W] \neq L$. To tackle this problem, Saito *et al.* [141] proposes a new estimator with propensity score as follow:

$$\hat{L}_{\text{sur}} = \frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \left[s_{ui} \left(\frac{1}{\rho_{ui}} \delta\left(\hat{r}_{ui}, 1\right) + (1 - \frac{1}{\rho_{ui}}) \delta\left(\hat{r}_{ui}, 0\right) \right) + (1 - s_{ui}) \delta\left(\hat{r}_{ui}, 0\right) \right] \tag{35}$$

where the propensity score $\rho_{ui} = p(e_{ui} = 1)$ is defined as the marginal probability of a user exposed to the item. $\hat{L}_{\text{sur}}$ is an unbiased estimator of the *True Risk*:

$$\mathbb{E}[\hat{L}_{\text{sur}}] = \mathbb{E}_{r_{ui}, e_{ui}} \left[\frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \left[e_{ui} r_{ui} \left(\frac{1}{\rho_{ui}} \delta\left(\hat{r}_{ui}, 1\right) + (1 - \frac{1}{\rho_{ui}}) \delta\left(\hat{r}_{ui}, 0\right) \right) + (1 - e_{ui} r_{ui}) \delta\left(\hat{r}_{ui}, 0\right) \right] \right]$$

$$= \mathbb{E}_{r_{ui}} \left[\frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \left[r_{ui} \delta\left(\hat{r}_{ui}, 1\right) + (1 - r_{ui}) \delta\left(\hat{r}_{ui}, 0\right) \right] \right] = \mathbb{E}_{r_{ui}} \left[\frac{1}{|\mathcal{U}||\mathcal{I}|} \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \delta\left(\hat{r}_{ui}, r_{ui}\right) \right] \tag{36}$$

$$= L$$

This propensity-based strategy is theoretical soundness and usually achieves better performance than weighting strategies. It is flexible and also has been extended to pair-wise objective function [138, 141, 165]. However, this kind of methods has some limitations: (1) its performance depends on the accuracy of the propensity score, which is quite challenging to obtain; (2) the inverse of propensity incurs high variance. Although these problems can be mitigated to a certain extent by some strategies (*e.g.,* joint learning [219], clapping [141]), they still deserve further exploration.

Causality-based Methods. Causal inference is another promising direction for addressing exposure bias. In fact, the spirit of a recommendation can be understood as to answer a counterfactual question: would the user interacts with the item if he had know the item? That is, we need to evaluate the causal estimand $p(S|do(E = 1), U, I)$ with intervening item exposure[5] rather than the statistical associations $p(S|U, V)$ estimated by vanilla recommender models. The intervention could remove the spurious association caused by exposure bias and recover users' true preference on the items.

Towards this target, various causality-based methods have been proposed. For example, Zhang *et al.*[208] resorted to the back-door criterion [132] to remove the exposure bias caused by the item popularity; Xu *et al.*[187] leveraged forward door criterion [132] to remove the effect from unobserved confounders; Wang *et al.*[168] leveraged counterfactual reasoning to eliminate the direct causal effect from exposure features on the prediction; Yang *et al.*[191] mitigates exposure bias through counterfactual samples; Liu *et al.*[108] disentangle the effect from exposure and preference with introducing information bottleneck.

Others. There are also some other strategies to address exposure bias in specific scenarios. Wang *et al.*[169] consider the queries of a search system are usually under-sampled to different extents, and thus are biased when click data is collected to learn the ranking function. They further propose a specific model for this situation, where queries are classified into different classes, and the bias in each class is estimated with randomized data. Ovaisi *et al.*[125, 126] attribute exposure bias to the fact that a user can examine only a truncated list of top-K recommended items. To address this kind of exposure bias, two-step Hechman method has been adopted. They first use a Probit model to estimate the probability of a document being observed and then leverage the exposure probability to correct the click model. Some recent work also consider users' sequential behaviors "exposure-click-conversion"

---

[5]Here the do-calculus $do(.)$ indicates the variable S is coercively intervened with a certain value. For more details on causal inference, we refer the readers to the excellent causal textbook [132].

and correspondingly devise an inverse propensity model [204], decomposition model [113, 181] or graph neural network [12] on the sequential behaviors to address exposure bias with multi-task learning. Besides, propensity scoring model has been utilized in debiasing explainable recommendation [42] and item-to-item recommendation [142].

4.4 Methods for Position Bias

Position bias is another type of bias that is widely studied in learning-to-rank systems, such as ad system and search engine. Position bias denotes that the higher ranked items will be more likely to be selected regardless of the relevance. Recent years have seen a number of work on position bias, and we categorize them into three lines.

Click Models. The first line is based on click models. The methods make hypotheses about user browsing behaviors and estimate true relevance feedback by optimizing the likelihood of the observed clicks. some work [26, 41, 48, 103, 205] on click models assume the examination hypothesis that if a displayed item is clicked, it must be both examined and relevant. This is based on the eye-tracking studies which testify that users are less likely to click items in lower ranks. To remedy position bias and to recover user true preference, they explicitly model the probability of an user clicks an item i at position q as follows:

$$
\begin{aligned}
&P(C = 1 \mid u, i, p) \\
&= \underbrace{P(C = 1 \mid u, i, E = 1)}_{r_{ui}} \cdot \underbrace{P(E = 1 \mid q)}_{h_q}
\end{aligned}
\tag{37}
$$

Notice that a hidden random variable E has been applied, which denotes whether the user has examined the item. In general, these methods make the following assumptions: if the user clicks it, the item must have been examined; if the user has examined the item, the click probability only depends on the relevance; and the examination depends solely on the position p. The model is highly similar to the exposure-based model for exposure bias except that the exposure probability is modeled with position.

Another choice of click model is the cascade model [41]. It differs from the above model in that it aggregates the clicks and skips in a single query session into a single model. It assumes a user examines an item from the first one to the last one, and the click depends on the relevance of all the items shown above. Let E_q, C_q be the probabilistic events indicating whether the q-th item is examined and clicked respectively. The cascade model generates users click data as follows:

$$
P(E_1) = 1 \tag{38}
$$

$$
P\left(E_{q+1} = 1 \mid E_q = 0\right) = 0 \tag{39}
$$

$$
P\left(E_{q+1} = 1 \mid E_q = 1, C_q\right) = 1 - C_q \tag{40}
$$

$$
P\left(C_q = 1 \mid E_q = 1\right) = r_{u_q, i} \tag{41}
$$

in which the Equation (40) implies that if a user finds her desired item, she immediately closes the session; otherwise she always continues the examination. The cascade model assumes that there is no more than one click in each query session, and if examined, an item is clicked with probability $r_{u_q, i}$ and skipped with $1 - r_{u_q, i}$. This basic cascade model has been further improved by considering the personalized transition probability [26, 59, 224]. Jin *et al.* [74] improve these models and consider users browsing behaviors in a more thorough manner with deep recurrent survival model.

However, these click models usually require a large quantity of clicks for each query-item or user-item pair, making them difficult to be applied in systems where click data is highly sparse, e.g., personal search [169]. Further, mis-specifying the generative process of users clicks will cause empirical bias and hurt recommendation performance.

Propensity Score. Another common solution to correct position bias is employing inverse propensity score, where each instance is weighted with a position-aware values [6]. The loss function is defined as follow:

$$L_{\text{IPS}}(f) = \sum_{u \in \mathcal{U}, i \in \mathcal{I}} \frac{1}{\rho(q)} s_{ui} \lambda(u, i | f) \tag{42}$$

Here we refer to [158] and use the ranking metrics. $\lambda(u, i|f)$ denotes the metric function that is based on the the rank of the item i for the user (or query) u according to the ranking system f. For instance, it can be chosen to match the well-known NDCG metric:

$$\lambda_{NDCG}(u, i | f) = \left(\log_2 \left(\text{rank} \left(i | u, f \right) + 1 \right) \right)^{-1} \tag{43}$$

A position-dependent propensity $\rho(q)$ is introduced to weight the λ function. The intuition behind the model is that clicks on items that are less likely to have been examined by users are weighted more heavily. This weighting strategy compensates for the effect of position bias on user exposure, allowing the method to estimate and learn without being affected by position bias in expectation Joachims *et al.* [77].

Estimating the propensity score for position bias have been well explored as its simplicity — just dependent on the item position. A simple yet effective solution to estimate a position-based propensity model is result randomization, where the ranking results are shuffled randomly and collect user clicks on different positions to compute propensities scores [67, 136, 144, 153, 169]. Because the expected item relevance is the same on all positions, it is provable that the difference of click rate on different positions produces an unbiased estimation of the truth propensities. Despite its simplicity and effectiveness, result randomization has a risk of significantly hurting the user experience as the highly ranking items may not be favored by the user. Pair-wise swapping [77] has been proposed to mitigate the problem, but can not eliminate negative effect completely. Therefore, the strategies that learn the propensity scores from the data without any intervention on the recommendation results have been explored. Fang *et al.* [52] and Agarwal *et al.* [8] adopt intervention harvesting, to learn the propensity. However, such methods require the feedback data from multiple ranking models. Further, some recent work [10, 77, 134, 171] consider learning a propensity model and a recommendation model as dual problem and develop specific EM algorithms to learn both models. More recently, the click model that captures the row skipping and slower decay phenomenon has been adopted to specify the propensity scores in [60], while cascade model has been adopted by [157]. Chen also *et al.*[35] propose to learn the propensity from the data observation.

Trust-aware Models. Item position not only influences users' exposure but also their decisions (*i.e.,* Both R and E are dependent on Q). Aforementioned propensity score is insufficient to address this problem. Hence, Agarwal *et al.*[7] propose an expansion to IPS to correct for both effects. The model hypothesizes that a real relevant item at position q can be misjudged with probability $1 - \epsilon_q^+$, while a non-relevant item can be clicked mistakenly with probability ϵ_q^-, *i.e.,* we have:

$$\begin{aligned} p(R = 1 | U, I, Q) &= p(r_{ui} = 1)\epsilon_q^+ + p(r_{ui} = 0)\epsilon_q^- \\ p(R = 0 | U, I, Q) &= p(r_{ui} = 1)(1 - \epsilon_q^+) + p(r_{ui} = 0)(1 - \epsilon_q^-) \end{aligned} \tag{44}$$

To tackle the label inversion caused by position bias, Agarwal *et al.*[7] extend IPS to the following objective:

$$L_{\text{Bayes-IPS}}(f) = \sum_{u,i} \frac{\epsilon_k^+}{\epsilon_k^+ + \epsilon_k^-} \frac{1}{\rho(q)} s_{ui} \lambda(u, i | f) \tag{45}$$

where an offset term $\epsilon_k^+/(\epsilon_k^+ + \epsilon_k^-)$ is introduced. Vardasbi *et al.*[158] further proof that $L_{\text{Bayes-IPS}}$ is insufficient and propose a more theoretical-soundness method with affinity corrections:

$$L_{\text{Affinity}}(f) = \sum_{u,i} \frac{s_{ui} - \rho(q)\epsilon_k^-}{\rho(q) \left(\epsilon_k^+ - \epsilon_k^- \right)} \lambda(u, i | f) \tag{46}$$

which is an unbiased estimation of the ideal estimator *w.r.t.* position bias.

4.5 Universal Solutions for Various Data Biases

Most existing methods are designed for addressing one or two biases of a specific scenario. Hence, when facing the real data that commonly contain multiple types of biases, these methods will fall short. Recently saw a few studies on universal solutions for multiple data biases and their combinations. These methods resorted to a small unbiased dataset for recommendation debiasing. For example, some work transfered the knowledge from the unbiased data to the target model with domain adaption [17, 106] or knowledge distillation [107]; More recently, Chen *et al.*[28] proposed to learn the optimal debiasing configures from the uniform data with meta learning.

Despite their effectiveness on handle various data biases, these methods require unbiased data, which is difficult and expensive to obtain. To collect uniform data, we must intervene in the system by using a random logging policy instead of the normal recommendation policy, which would hurt users' experience and revenues of the platform. Therefore, how to develop a universal solution without using unbiased data is still an open problem and deserves further exploration.

4.6 Methods for Popularity Bias

Popularity bias is a common problem in recommendation systems. We categorize the methods into four types.

Regularization. Suitable regularization can push the model towards balanced recommendation lists. Abdollahpouri *et al.* [2] introduced *LapDQ* regularizer [179] $tr(Q^T L_D Q)$, where Q denotes the item embedding matrix, $tr(\cdot)$ denotes the trace of a matrix, and L_D denotes the Laplacian matrix of D, where $D_{i,j} = 1$ if item i and j belong to the same set (popular items or long-tail items) and 0 otherwise. Kamishima *et al.* [83] applied the *mean-match* regularizer [82] in their *information-neutral recommender systems (INRS)* to correct popularity bias. They first introduced mutual information to measure the influence of features on the recommendation results, and through a series of mathematical approximations and derivations, they obtain a specific regularization term: $-(M_{D^{(0)}}(\{\hat{r}\}) - M_{D^{(1)}}(\{\hat{r}\}))^2$, where $M_D(\{\hat{r}\}) = \frac{1}{|D|} \sum_{(x_i,y_i,v_i) \in D} \hat{r}(x_i, y_i, v_i)$. More recently, Zhu *et al.* [221] utilized a *Pearson Coefficient* regularizer to decrease the correlation between item popularity and model output scores. Note that the above regularizers are result-oriented, guiding the model to give more balanced results.

Different from result-oriented regularizers, Chen *et al.* [38] devise a process-oriented regularization term. It attributes the inability of effectively recommending long-tail items as the insufficient training of them. These items usually have few interaction records and thus their embedding vectors can not be well trained, making their prediction scores close to the initial values and remain neutral. Motivated by this point, Chen *et al.* proposed *Entire Space Adaptation Model (ESAM)* from the perspective of domain adaptation (DA). ESAM aims to transfer the knowledge from these well-trained popular items to the long-tail items. ESAM introduced three regularization terms for transferring as: (1) Domain adaptation with item embedding (or attributes) correlation alignment: $L_{DA} = \frac{1}{L^2} \sum_{(j,k)} (h_s^{j^T} h_s^k - h_t^{j^T} h_t^k)^2 = \frac{1}{L^2} ||Cov(D^s) - Cov(D^t)||_F^2$, where $|| \cdot ||_F^2$ denotes squared matrix Frobenius norm. $Cov(D^s) \in R^{L*L}$ and $Cov(D^s) \in R^{L*L}$ represent the covariance matrices of high-level item attributes, which can be computed as $Cov(D^s) = D^{s^T} D^s$, and $Cov(D^t) = D^{t^T} D^t$. s means source domain (popular items), and t means target domain (unpopular items). (2) Center-wise clustering for source clustering L_{DC}^c: encouraging the features of the items with the same feedback (such as buy, view, and click) to be close together, and the features of the items with different feedbacks to move away from each other. (3) Self-training for target clustering L_{DC}^p: minimizing the entropy regularization $-p\log p$ favors a low-density separation between classes. This term is a way of self-training which increases the discriminative power between non-displayed items.

Adversarial Learning. Adversarial learning is another line to address popularity bias*et al.*[91]. The basic idea is to play a min-max game between the recommender G and the introduced adversary D, so that D gives a signal to improve the recommendation opportunity of the niche items. In [91], The adversary D takes the synthetically

generated popular-niche item pairs $(\tilde{i}^p, \tilde{i}^n | u)$, and an equal number of true popular-niche pairs (i^p, i^n) as input. True pairs (i^p, i^n) are sampled from their global co-occurrence and synthetic pairs $(\tilde{i}^p, \tilde{i}^n)$ are drawn by the recommender. The recommender G can be instantiated with recent recommendation model such as NCF. Through adversarial learning between G and D, D learns the implicit association between popular and niche items, while G learns to capture more niche items that correlate with the user's history, resulting in recommending more long-tail items for users.

Causal Graphs. Causal graph is a powerful tool for counterfactual reasoning. Some recent work proposed to leverage causal graph to tackle popularity bias. They first built a causal graph to elucidate popularity bias, and then applied counterfactual intervention over the graph to mitigate the bias. For example, Zhang [208] *et al.* attributed the popularity bias to the undesirable causal effect from item popularity to the item exposure. To eliminate this effect, they further proposed to intervene the distribution of the exposed items with back-door criterion or propensity score; Zhao *et al.* [215] and Wang *et al.* [217] traced popularity bias from conformity effect (*i.e.,* the effect of item popularity on user behavior), and causally intervened the item popularity to make fair recommendation; Analogically, Wei *et al.* [180] performed counterfactual reasoning to eliminate the direct effect of item (popularity) to the prediction; Wang *et al.* [167] studied how popularity bias occurs in model training. They attributed the popularity bias to a confounding causal structure and applied backdoor adjustment to mitigate this effect.

Others. There are some other methods on popularity bias. one solution to reduce popularity bias is through introducing other side information. For example, Bressan *et al.* leverage social information to reduce popularity bias [19]. Abdollahpouri gives a different strategy [1], which relies on re-ranking. To perform top-k recommendation, it first generates a relatively large recommendation list with a classical model, and then re-ranks the list by considering the item popularity. Similar to exposure bias, propensity score can also be applied to reduce popularity bias: by decreasing the influence of popularity items to model training, the popularity bias can be mitigated [190].

4.7 Methods for Unfairness

Before introducing existing fairness-aware methods, we first give some formulations of fairness.

4.7.1 Fairness Formulations. There are extensive studies on fairness in machine learning. Without loss of generality, we use the notation of prediction model throughout this section to discuss fairness. Let A, X, U be the set of sensitive attributes (*aka.* protected attributes), other observed attributes, and unobserved attributes of an individual, respectively. Y denotes the ground-truth outcome to be predicted, while $\hat{Y}$ is the prediction produced by a prediction model that depends on A, X, U. For simplicity we often assume A is encoded as a binary attribute, but this can be generalized to other cases.

There exist many different variations of fairness definition, which can be roughly categorized into four types: 1) **fairness through unawareness** [58]; 2) **individual fairness** [49, 79, 111, 200]; 3) **group fairness** (*e.g.*, demographic parity [86, 198], equality of opportunity [61, 197], predictive equality [39], equalized odds [61], calibration within groups [88]); and 4) **counterfactual fairness** [93]. Here we present some widely-used formulations:

- **Fairness Through Unawareness**: *A model is fair if any sensitive attributes A are not explicitly used in the modeling process.*
- **Individual Fairness**: *A model is fair if it gives similar predictions to similar individuals. Formally, if individuals i and j are similar under a certain metric, their predictions should be similar:* $\hat{Y}(X(i), A(i)) \approx \hat{Y}(X(j), A(j))$.
- **Demographic Parity**: *Each protected group (i.e., with the same sensitive attributes) should receive positive prediction at an equal rate. Formally, the prediction* $\hat{Y}$ *satisfies demographic parity if* $P(\hat{Y}|A = 0) = P(\hat{Y}|A = 1)$.

- **Equality of Opportunity**: *Given the prediction model, the likelihood of being in the positive class is the same for each protected group. Formally, the prediction $\hat{Y}$ satisfies the equality of opportunity if $P(\hat{Y} = 1|A = 0, Y = 1) = P(\hat{Y} = 1|A = 1, Y = 1)$.*
- **Counterfactual fairness**: *Given a causal model $(U, A \cup X, F)$, the prediction $\hat{Y}$ is counterfactually fair if under any context $X = x$ and $A = a$, $P(\hat{Y}_{A \leftarrow a}(U) = y|X = x, A = a) = P(\hat{Y}_{A \leftarrow a'}(U) = y|X = x, A = a)$, for all y and for any value a' attainable by A.*

Besides these general definitions *w.r.t.* user attributes, the concept of fairness has been generalized to multiple dimensions in recommender systems [20], spanning from fairness-aware ranking [11, 15, 57], fairness in terms of user psychological characteristics [164], supplier fairness in two-sided marketplace platforms [120], provider-side fairness to make items from different providers have a fair chance of being recommended [83, 109], fairness in group recommendation to minimize the unfairness between group members [105].

4.7.2 Fairness-aware Methods. In the following, we review four different ways to mitigate the unfairness issue on recommendation.

Rebalancing. Inspired by the strategy used to tackle the class-imbalance problem, one common paradigm is to balance the data or recommendation results *w.r.t.* certain fairness target like demographic parity. Some intensively-adopted strategies in machine learning research are re-labeling the training data to make the proportion of positive labels equal in the protected and unprotected groups [133], or re-sampling the training data to achieve statistical parity [57].

This idea of rebalancing data is prevalent in fairness-aware ranking, where the fairness constraint can be represented in various forms. Towards individual equity-to-attention fairness in rankings, previous work [11, 15] propose multiple ranking functions to sort items and then achieve fairness amortized across these rankings. Towards group fairness, FA∗IR [199] is a post-processing method to achieve fair top-K ranking *w.r.t.* group fairness criteria, in which a subset of K candidates are re-selected from a large item collection to achieve a required proportion for a single under-represented group. Analogously, DetCons and DetConstSort [57] formalize the fairness as a desired distribution over sensitive attributes, and re-rank candidates (*i.e.,* LinkedIn users) to satisfy the constraints; Li *et al.* [98] constrain the difference of the average recommendation performance between two groups, and formulate the fairness-aware ranking problem as 0-1 integer programming. To formulate group fairness in terms of exposure allocation, Singh *et al.* [146] propose a framework for formulating fairness constraints on rankings, and sample rankings from an associated probabilistic algorithm to fulfill the constraints. HyPER [53] uses probabilistic soft logic (PSL) rules to balance the ratings for both users in protected and unprotected groups, where fairness constraints are encoded as a set of rules. More recently, when organizing user-item interactions in the form of graph, some work [22, 135] study potential unfairness issue inherent within graph embedding. Among them, Fairwalk [135] treats the group information *w.r.t.* sensitive attributes as a prior distribution, and then performs node2vec based on the prior to sample random walks and generate debiased embeddings, which are evaluated in friendship recommendation.

Regularization. The basic idea of the regularization line is to formulate the fairness criteria as a regularizer to guide the optimization of model. A general framework, *Learned Fair Representation (LFR)*, is proposed in [200], which generates the data representations to encode insensitive attributes of data, while simultaneously removing any information about sensitive attributes *w.r.t.* the protected subgroup. Formally, it is composed of three loss components:

$$\min \mathcal{L} = \alpha C(X, R) + \beta D(R, A) + \gamma E(Y, R) \tag{47}$$

where $C(\cdot)$ is the reconstruction loss between input data X and representations $R = Enc(X)$ with an encoder function $Enc(\cdot)$; $E(\cdot)$ is the prediction error in generating prediction Y from R, such as cross entropy; $D(\cdot)$ is a regularization term that measures the dependence between R and sensitive attribute A, which is defined as

fairness constraints such as demographic parity:

$$D(R, A) = |\mathbb{E}_R P(R|A = 1) - \mathbb{E}_R P(R|A = 0)| \tag{48}$$

where $P(R|A = 1)$ relies on the distance of representation R and the centroid representation $\tilde{R}_1$ of the group where $A = 1$:

$$P(R|A = 1) = \frac{\exp -||R - \tilde{R}_1||_2}{\sum_{a \in \{0,1\}} \exp -||R - \tilde{R}_a||_2} \tag{49}$$

Using such a regularization makes the encoded representation sanitized and blind to whether or not the individual X is from the protected group.

Studies on this research line have been extensively conducted by subsuming different fairness formulations under the foregoing framework. Earlier, Kamishima *et al.* first claimed the importance of neutrality (*aka.* viewpoint of sensitive attribute) in recommendation [81], and then proposed two methods — (1) one regularization-based matrix completion method [82], where the fairness regularizer is formulated as the negative mutual information $-I(A; Y)$ between sensitive attribute A and prediction Y, and (2) one graphical model-based method [84], where the fairness regularizer accounts for the expected degree of independence between A and Y in the graphical model. Later, Kamishima *et al.* generalized these work to implicit feedback-based recommender systems [80]. Analogously, Yao *et al.* [192, 193] proposed four fairness metrics in collaborative filtering, and used similar regularization-based optimization method to mitigate different forms of bias.

Moreover, there are some regularization-based studies working on more specific scenarios. For example, Abdollahpouri *et al.* [2] focused on controlling popularity bias in learning-to-rank recommendation, and proposed a regularizer that measures the lack of fairness for the short-head and medium-tail item sets in a given recommendation list to improve fairness during model training. Xiao *et al.* [105] worked on fairness-aware group recommendation, and designed a multi-objective optimization model to minimize the utility gap between group members. Burke *et al.* [21] proposed a regularization-based matrix completion method to reweigh different neighbors, in order to balance the fairness between protected and unprotected neighbors in collaborative recommendation. Zhu *et al.* [222] presented a fairness-aware tensor-based recommendation approach, which uses sensitive latent factor matrix to isolate sensitive features and then uses a regularizer to extract sensitive information which taints other factors. More recently, going beyond the pointwise fairness metrics in ranking, Beutel *et al.* [14] considered pairwise fairness of user preference between clicked and unclicked items, and offered a new regularizer to encourage improving this metric.

Besides in optimization objective, regularization also has been added in the ranking policy to address the unfairness issue. [121] considers the problem in dynamic ranking system, where the ranking function dynamically evolves based on the feedback that users provide, and present a new sorting criterion FairCo as follows:

$$\sigma_\tau = \underset{i \in I}{\text{argsort}} \left(\hat{R}(i \mid u) + \lambda \, \text{err}_\tau(i) \right) \tag{50}$$

where the error term err measures the fairness violation has been introduced. The intuition behind FairCo is that the error term pushes the items from the underexposed groups upwards in the ranking lists. Fairness-aware constraints have also been introduced in learning-to-rank (LTR) models [147, 188]. Efficient policy-gradient algorithms have been developed for model optimization [124, 188].

Adversarial Learning. Similar with the idea of LFR (*cf.* (47)), the line of adversarial learning aims to get fairness as a side-effect of fair representation. The basic idea is to play a min-max game between the prediction model and an adversary model, where the adversary tries to predict the sensitive attributes from the data representations, so minimizing the performance of the adversary is to remove the information pertinent to the sensitive attributes in the representation. Towards this goal, a general framework, *Adversarial Learned Fair*
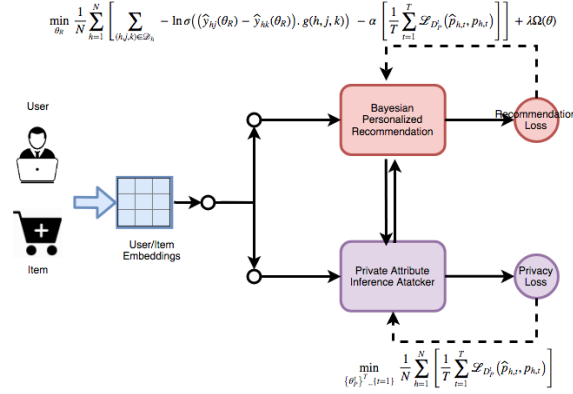
$$\min_{\theta_R} \frac{1}{N} \sum_{h=1}^{N} \left[\sum_{(h,j,k) \in \mathcal{D}_h} -\ln \sigma \left((\hat{y}_{hj}(\theta_R) - \hat{y}_{hk}(\theta_R)) \cdot g(h,j,k) \right) - \alpha \left[\frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{D'_{\hat{p}}}(\hat{p}_{h,t}, p_{h,t}) \right] \right] + \lambda \Omega(\theta)$$

$$\min_{\{\theta'_p\}^T\_{\{t=1\}}} \frac{1}{N} \sum_{h=1}^{N} \left[\frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_{D'_{\hat{p}}}(\hat{p}_{h,t}, p_{h,t}) \right]$$

Fig. 8. Recommendation with attribute protection (RAP) [13]. The figure was reproduced from the Arxiv[6] with authors' permission.

Representation (ALFR), is proposed in [50] which is formulated as follows:

$$\max_{\phi} \min_{\theta} \mathcal{L} = \alpha C_{\theta}(X, R) + \beta D_{\theta,\phi}(R, A) + \gamma E_{\theta}(Y, R) \tag{51}$$

where $C_{\theta}(\cdot)$ is the reconstruction loss to quantify the information retained in the representations R about the data X by the ability of an encoder or decoder network; $E_{\theta}(\cdot)$ is to predict Y from R via a predictor network; θ encompasses the parameters of the encoder/decoder and predictor networks; and $D_{\theta,\phi}(\cdot)$ is to quantify the independence between the representation R and the sensitive attributes A via an adversary network: $R \rightarrow A$. Assuming A is binary, $D_{\theta,\phi}(\cdot)$ is formulated as log-loss for binary adversary network f:

$$D = \mathbb{E}_{X,A} A \cdot \log(f(R)) + (1 - A) \cdot \log(1 - f(R)) \tag{52}$$

which satisfies the fairness constraint of demographic parity. Maximizing $D_{\theta,\phi}(\cdot)$ is to optimize the adversary's parameters ϕ, while minimizing $D_{\theta,\phi}(\cdot)$ is to optimize the representation parameters θ.

Only recently have researchers considered this line in the field of recommendation. For example, Zhu *et al.* [223] leveraged adversarial learning to enhance the score distribution similarity between different groups. Bose *et al.* [18] and Wu *et al.* [183] extended the ALFR framework by enforcing compositional fairness constraints on graph embeddings for multiple sensitive attributes, which are evaluated in the scenarios of item or friendship recommendation. Wherein, instead of fair *w.r.t.* single sensitive attribute, it makes the graph embeddings be invariant *w.r.t.* different combinations of sensitive attributes by employing a compositional encoder in the adversary network. Building upon the ALFR framework, Beigi *et al.* [13] proposed a framework termed *recommendation with attribute protection (RAP)* to recommend items based on user preference, while simultaneously defensing against private-attribute inference attacks. In particular, the prediction and adversarial networks are instantiated as the sensitive attribute inference attacker and the Bayesian personalized recommender, respectively. Analogically, Li *et al.* [99] targeted at personalized counterfactual fairness with leveraging adversarial learning to isolated the personalized sensitive attributes.

Causal Modeling. Inspired by the success of causal modeling [131], studying fairness from the causal perspective [93, 122, 185, 186, 202] has attracted increasing attentions. In general, fairness is formulated as the causal effect of the sensitive attribute, which is evaluated by applying counterfactual interventions over a causal graph. For example, Wu *et al.* [185] focused on fairness-aware ranking, and argued that the fairness constraints based

---

[6]https://arxiv.org/pdf/1911.09872.pdf

on statistical parity hardly measure the discriminatory effect. Hence, they built a causal graph that consists of the discrete profile attributes and the continuous score, and proposed a path-specific effect technique to detect and remove both direct and indirect rank bias. Kusner *et al.* [93] introduced the notion of counterfactual fairness, which is derived from Pearl's causal model [131]. It considers the causal effect by evaluating the counterfactual intervention — more formally, for a particular individual, whether its prediction in the real world is identical to that in the counterfactual world where the individual's sensitive attributes had been different.

Others. There are some other strategies on unfairness. For example, Li *et al.* [97] proposed to add an autoencoder layer when learning user and item representation. This treatment can enforce that the specific unique properties of all users and items are sufficiently preserved in the representation, mitigating the bias towards mainstream users. Islam *et al.* [69] first computed a group-specific bias direction, and then debiased each user representation by subtracting its component in that direction. Ge *et al.* [55] studied on the problem of long-term fairness and proposed a fairness-constrained reinforcement learning algorithm to adapt dynamic fairness requirement.

4.8 Methods for Mitigating Loop Effect

Practise recommender systems usually create a pernicious feedback loop, which will create bias and further intensify bias over time. To deal with this problem, besides the aforementioned strategies on a specific bias, a surge of methods have been proposed recently to reduce the iterated bias that occurs during the successive interaction between users and recommender system.

Uniform data. Leveraging uniform data is the most straightforward way to address the problem. To collect uniform data, this kind of methods intervene in the system by using a random logging policy instead of a normal recommendation policy. That is, for each user, they do not use the recommendation model for item delivery, but instead randomly select some items and rank them with a uniform distribution [73, 107]. The uniform data often provide gold-standard unbiased information because it breaks the feedback loop and is not affected by various biases. However, the uniform policy would inevitably hurt users' experience and the revenue of the platform, thus it is usually restricted to a small percentage of online traffic. Therefore, how to correct the bias with a small uniform data is a key research question. Yuan *et al.*[196] learn a imputation model from the uniform data and apply the model to impute the labels of all displayed or non-displayed items. Rosenfeld *et al.* [139] and Bonner *et al.*[17] employ two recommendation models for the biased data and uniform data, and further use a regularization term to transfer the knowledge between the models; Liu *et al.* [107] leverage knowledge distillation to extract information from uniform data to learn a unbiased recommendation model. Yu *et al.* [195] leverage influence function to reweight training instances so that it has less loss in an unbiased validation set. Chen *et al.* [28] proposed to learn the optimal debiasing configures from the unbiased data.

Reinforcement learning. Collecting uniform data with a random policy is not a satisfactory strategy as it hurts recommendation performance. Smarter recommendation strategy or policy needs to be explored. There exists an exploration-exploitation dilemma in recommender system, where the exploitation is to recommend items that are predicted to best match users' preference, while the exploration is to recommend items randomly to collect more unbiased user feedback to better capture user preference. To deal with this problem, a large number of work explores interactive recommendation by building a reinforcement learning (RL) agent. Figure 9 illustrates the system-user interactions with a RL agent. Different from traditional recommendation methods, RL considers the information seeking tasks as sequential interactions between an RL agent (system) and users (environment). During the interaction, the agent can continuously update its strategies π according to users' history information or feedback (i.e. state s_t) and generates a list of items (i.e. action a_t) that best match users' preferences or explore users' preference for long term reward. Then, the users will give the feedback (i.e. rewards r_t, such as ratings or clicks) on the recommendation lists to update the agent. Therefore, RL could balance the competition between the exploitation and exploration and maximize each user's long term satisfaction with
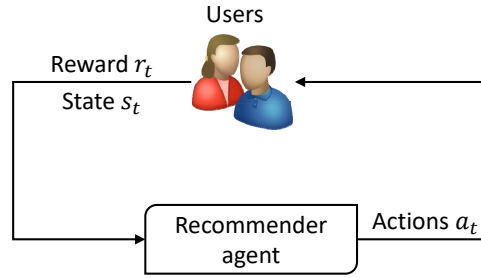
Fig. 9. The system-user interactions with a RL aggent. The figure is plotted referring to [211] with permission.

the system [209]. Some recent work [96, 161, 173, 213] balance exploitation and exploration in bandit setting with ε-greedy, Boltzmann Exploration or Upper Confidence Bounds (UCB). Some work estimates action-value reward function $Q(s, a)$ with Q network using the Bellman equation and finds the best strategy with the largest function value [36, 211, 214, 216]. Also, the actor network has been adopted recently to learn the best policy by maximizing the long term reward [27, 170, 210, 212].

A challenge of RL-based recommender is how to evaluate a policy. It is best to deploy it online, e.g., in the form of an A/B test, which however is expensive and time-consuming in terms of engineering and logistic overhead and also may harm the user experience when the policy is not mature [70]. Off-policy evaluation is an alternative strategy that uses historical interaction data to estimate the performance of a new policy. However, off-policy evaluation will suffer from bias as the data are collected by an existing biased logging policy instead of uniform policy. To correct the data bias, Chen *et al.* [34] proposes to weight the policy gradient with the inverse of the probability of historical policy. Inspired by [34], some work [70, 117, 154] further explore off-policy evaluation for non-stationary recommendation environments or slate recommendation. However, as claimed by Jeunen *et al.*[72], existing off-policy learning methods usually fail due to stochastic and sparse rewards. Therefore, they [72] further propose to leverage supervised signal with IPS strategy to better evaluate a policy. Nevertheless, off-policy evaluation is still a challenging task especially when the historical policy is not provided, which deserves for further exploration.

Others. There are some other strategies to mitigate the loop effect. Sun *et al.*[152] leverage blind spot term to let items be close to each other in the latent space. Sinha *et al.*[148] provide an algorithm for deconvolving feedback loops to recover users' truth rating values.

5 FUTURE WORK

This section discusses open issues and point out some future directions.

5.1 Evaluation of Propensity Scores

As mentioned before, Inverse Propensity Score is a conventional strategy to debias. However, the effectiveness and unbiasedness of an IPS strategy are guaranteed only when the propensity scores are properly specified. How to obtain proper propensity scores remains an important research question. Existing methods usually assume the ideal propensities are given. Although the evaluation of propensity scores in some simple scenarios, e.g. for position bias, have been explored, evaluating propensity scores in more complex scenarios, such as for selection bias or exposure bias, is still an open problem and deserves further exploration.

5.2 General Debiasing Framework

From former studies, we can find that existing methods are usually designed for just addressing one or two specific biases. However, in the real world, various biases usually occur simultaneously. For example, users usually rate the items that they like and their rating values are influenced by the public opinions, where conformity bias and selection bias are mixed in the collected data. Besides, the distribution of rated user-item pairs is usually inclined to popular items or specific users groups, making the recommendation results easily suffer from popularity bias and unfairness. It is imperative that recommender systems require a general debiasing framework to handle the mixture of biases. It is a promising but largely under-explored area where more studies are expected. Although challenging, the simple case — the mixture of just two or three biases — is worth to be explored first.

IPS or its variants, which have been successfully applied for various biases, are a promising solution for this problem. It will be interesting and valuable to explore a novel IPS-based framework, which summarizes the applications of IPS on different kinds of biases and provides a general propensity score learning algorithm.

5.3 Better Evaluation

How to evaluate a recommender system in an unbiased manner? It is an essential question for both researchers and practitioners in this area. Existing methods either require accurate propensity scores or rely on a considerable amount of unbiased data. However, the accuracy of the former can not be guaranteed, while the latter hurts user experience and is usually constrained on a very small percentage of online traffic. Uniform data provides gold-standard unbiased information but its small scale makes it insufficient to thoroughly evaluate a recommendation model due to high variance. Exploring new evaluators using large-scale biased data and small-size unbiased data will be an interesting direction. More theoretical studies are expected, analyzing the expectation, bounds and confidences of the proposed evaluator.

Due to popularity bias and unfairness, the evaluation exhibits more difficulties. Different work usually adopts different evaluation criteria of popularity bias or unfairness. This creates an inconsistent reporting of scores, with each author reporting their own assortment of results. The performance or comparisons of existing methods can not be well understood. As such, we believe that a suite of benchmark datasets and standard evaluations metrics should be proposed.

5.4 Knowlege-enhanced Debiasing

It is natural that exploiting the abundant auxiliary information would improve the efficacy of debiasing. Recent years have seen some examples that leverage attributes of users or items to correct biases in recommendation. An interesting direction is how to better exploit this auxiliary information as the attributes are not isolated but connected with each other forming a knowledge graph. The knowledge graph captures much more rich information, which could be useful to understand the data bias. For example, given a user u_1 watches movies i_1 and i_2, both of which are directed by the same person p_1 and of the same genre p_2. From the knowledge graph, we can deduce that the u_1 are highly likely to have known the movies that connect with entities i_1, i_2, p_1 or p_2. This exposure information is important for exposure bias correction. Another advantage of knowledge graph is its generality. All data, data sources, and databases of every type can be represented and operationalized by the knowledge graph. Knowledge graph would be a powerful tool for developing a feature-enhanced general debiasing framework.

5.5 Explanation and Reasoning with Causal Graph

Cause graph is an effective mathematical tool for elucidating potentially causal relationships from data, deriving causal relationships from combinations of knowledge and data, predicting the effects of actions, and evaluating explanations for observed events and scenarios. As such, it is highly promising for the debiasing tasks in

recommendation. On the one hand, the key of debaising is to reason the occurrence, cause, and effect over recommendation models or data. Most biases can be understood with mild cause assumptions and additional confounding factors in the causal graph. The effect of bias also can be inferred through the casual paths in the graph. On the other hand, recommendation is usually considered as an intervention analogous to treating a patient with a specific drug, where counterfactual reasoning needs to be conducted. What happens if the recommended items are exposed to the users? Causal graph provides potentials to answer this question. The formulated unbiased recommendation criteria can be derived with causal graph.

Nowadays, making explainable recommendations is increasingly important as it helps to improve the transparency, persuasiveness, effectiveness, trustworthiness, and satisfaction of a RS. Explainable recommendation and debiasing are highly related in the sense that they both address the problem of why: they both need to answer why certain items are recommended by the algorithm. When causal graph is promising to address the bias problem in a RS, it can also provide opportunities to give explanation from the strong causal paths in the graph.

To this end, the next step would to be to design better and suitable causal graph, which is capable of reasoning, debiasing, and explanation. We believe causal model will bring the recommendation research into a new frontier.

5.6 Dynamic Bias

In real world, biases are usually dynamic rather than static. For example, the fashion of clothes changes frequently; users experience many new items and may get new friends every days; the recommendation system will update its recommendation strategy periodically; etc. All in all, factors or biases often evolve with the time going by. It will be interesting and valuable to explore how bias evolves and analyze how the dynamic bias affects a RS.

5.7 Double-edged Sword of Bias

Bias is not always harmful. For example, popularity bias has been validated as a double-edged sword [208, 215]. Popularity bias not only results from conformity but also item quality. Appropriately leveraging popularity bias in recommendation may improve the performance. It will be interesting and valuable to explore the double-edged nature of other biases, fostering their benign effects while circumventing their harmful.

5.8 Fairness-Accuracy Trade-off

The trade-off between accuracy and fairness is of importance in recommendation scenarios, where equally treating different groups *w.r.t.* sensitive attributes has been shown to sacrifice the recommendation performance. Hence, it inspires us to (1) identify specific unfairness issues; and (2) define the fairness criteria carefully to cover a wide range of use cases; and 3) design some controllable methods, where the trade-off between fairness and accuracy can be controlled. Moreover, existing methods largely assume that the sensitive attributes of users (or items, groups) are provided as part of the input. Such assumptions might not hold in certain real-world scenarios — for example, in collaborative filtering, user profiles including sensitive attributes like age and gender cause different patterns of their behaviors; however, such profiles are unobserved but implicitly affect the recommendation performance. A research direction is to understand the dimensions of causality and design fairness-aware collaborative filtering algorithms in case sensitive attributes are not readily available.

6 CONCLUSIONS

In this article, with reviewing more than 180 papers, we systematically summarize the seven kinds of biases in recommendation, along with providing their definitions and characteristics. We further devise a taxonomy to organize and position existing debiasing approaches, with discussing their strengths and weaknesses. We list some open problems and research topics worth to be further explored. We hope this survey can benefit

the researchers and practitioners who are keen to understand the biases in recommendation and inspire more research work in this area.

REFERENCES

[1] Himan Abdollahpouri. 2019. Popularity Bias in Ranking and Recommendation. In *AIES*. 529–530.

[2] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Controlling Popularity Bias in Learning-to-Rank Recommendation. In *RecSys*. 42–46.

[3] Himan Abdollahpouri and Masoud Mansoury. 2020. Multi-sided exposure bias in recommendation. *arXiv preprint arXiv:2006.15772* (2020).

[4] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286* (2019).

[5] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2020. The connection between popularity bias, calibration, and fairness in recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 726–731.

[6] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A general framework for counterfactual learning-to-rank. In *SIGIR*. 5–14.

[7] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. 2019. Addressing trust bias for unbiased learning-to-rank. In *The World Wide Web Conference*. 4–14.

[8] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *WSDM*. 474–482.

[9] Charu C Aggarwal. 2016. Neighborhood-based collaborative filtering. In *Recommender systems*. Springer, 29–70.

[10] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *SIGIR*. 385–394.

[11] Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing Fair Ranking Schemes. In *SIGMOD*. 1259–1276.

[12] Wentian Bao, Hong Wen, Sha Li, Xiao-Yang Liu, Quan Lin, and Keping Yang. 2020. GMCM: Graph-based Micro-behavior Conversion Model for Post-click Conversion Rate Estimation. In *SIGIR*. 2201–2210.

[13] Ghazaleh Beigi, Ahmadreza Mosallanezhad, Ruocheng Guo, Hamidreza Alvari, Alexander Nou, and Huan Liu. 2020. Privacy-Aware Recommendation with Private-Attribute Protection using Adversarial Learning. In *WSDM*. 34–42.

[14] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H. Chi, and Cristos Goodrow. 2019. Fairness in Recommendation Ranking through Pairwise Comparisons. In *KDD*. 2212–2220.

[15] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *SIGIR*. 405–414.

[16] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (Technology) is Power: A Critical Survey of" Bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. 5454–5476.

[17] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *RecSys*. 104–112.

[18] Avishek Joey Bose and William L. Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *ICML*, Vol. 97. 715–724.

[19] Marco Bressan, Stefano Leucci, Alessandro Panconesi, Prabhakar Raghavan, and Erisa Terolli. 2016. The Limits of Popularity-Based Recommendations, and the Role of Social Ties. In *KDD*. 745–754.

[20] Robin Burke. 2017. Multisided Fairness for Recommendation. *CoRR* abs/1707.00093 (2017).

[21] Robin Burke, Nasim Sonboli, Masoud Mansoury, and Aldo Ordoñez-Gauger. 2017. Balanced Neighborhoods for Fairness-aware Collaborative Recommendation. In *FATREC Workshop on Responsible Recommendation Proceedings*. 5.

[22] Maarten Buyl and Tijl De Bie. 2020. DeBayes: a Bayesian method for debiasing network embeddings. *CoRR* abs/2002.11442 (2020).

[23] Rocío Cañamares and Pablo Castells. 2018. Should i follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems. In *SIGIR*. ACM, 415–424.

[24] Allison JB Chaney, David M Blei, and Tina Eliassi-Rad. 2015. A probabilistic model for using social networks in personalized item recommendation. In *RecSys*. ACM, 43–50.

[25] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *RecSys*. 224–232.

[26] Olivier Chapelle and Ya Zhang. 2009. A dynamic bayesian network click model for web search ranking. In *WWW*. 1–10.

[27] Haokun Chen, Xinyi Dai, Han Cai, Weinan Zhang, Xuejian Wang, Ruiming Tang, Yuzhou Zhang, and Yong Yu. 2019. Large-scale interactive recommendation with tree-structured policy gradient. In *AAAI*, Vol. 33. 3312–3320.

[28] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to Debias for Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information*

Retrieval, Virtual Event, Canada, July 11-15, 2021. ACM, 21–30.

[29] Jiawei Chen, Yan Feng, Martin Ester, Sheng Zhou, Chun Chen, and Can Wang. 2018. Modeling Users' Exposure with Social Knowledge Influence and Consumption Influence for Recommendation. In *CIKM*. 953–962.

[30] Jiawei Chen, Can Wang, Martin Ester, Qihao Shi, Yan Feng, and Chun Chen. 2018. Social recommendation with missing not at random data. In *ICDM*. IEEE, 29–38.

[31] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. 2020. Fast Adaptively Weighted Matrix Factorization for Recommendation with Implicit Feedback.. In *AAAI*. 3470–3477.

[32] Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2019. SamWalker: Social Recommendation with Informative Sampling Strategy. In *The World Wide Web Conference*. ACM, 228–239.

[33] Jiawei Chen, Xiang Wang, Fuli Feng, and Xiangnan He. 2021. Bias Issues and Solutions in Recommender System: Tutorial on the RecSys 2021. In *Fifteenth ACM Conference on Recommender Systems*. 825–827.

[34] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender system. In *WSDM*. 456–464.

[35] Mouxiang Chen, Chenghao Liu, Jianling Sun, and Steven CH Hoi. 2021. Adapting Interactional Observation Embedding for Counterfactual Learning to Rank. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.

[36] Shi-Yong Chen, Yang Yu, Qing Da, Jun Tan, Hai-Kuan Huang, and Hai-Hong Tang. 2018. Stabilizing reinforcement learning in dynamic environment with application to online recommendation. In *KDD*. 1187–1196.

[37] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *KDD*. 785–794.

[38] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. ESAM: Discriminative Domain Adaptation with Non-Displayed Items to Improve Long-Tail Performance. In *SIGIR*. 579–588.

[39] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163.

[40] Andrew Collins, Dominika Tkaczyk, Akiko Aizawa, and Joeran Beel. 2018. A Study of Position Bias in Digital Library Recommender Systems. *arXiv preprint arXiv:1802.06565* (2018).

[41] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *WSDM*. 87–94.

[42] Khalil Damak, Sami Khenissi, and Olfa Nasraoui. 2021. Debiased Explainable Pairwise Ranking from Implicit Feedback. In *Fifteenth ACM Conference on Recommender Systems*. 321–331.

[43] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. *Proceedings on privacy enhancing technologies* 2015, 1 (2015), 92–112.

[44] Eustasio del Barrio, Paula Gordaliza, and Jean-Michel Loubes. 2020. Review of Mathematical frameworks for Fairness in Machine Learning. *arXiv preprint arXiv:2005.13755* (2020).

[45] Robin Devooght, Nicolas Kourtellis, and Amin Mantrach. 2015. Dynamic matrix factorization with priors on unknown values. In *KDD*. 189–198.

[46] Jingtao Ding, Fuli Feng, Xiangnan He, Guanghui Yu, Yong Li, and Depeng Jin. 2018. An improved sampler for bayesian personalized ranking by leveraging view data. In *Companion of the The Web Conference 2018 on The Web Conference 2018*. IW3C2, 13–14.

[47] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced negative sampling for recommendation with exposure data. In *IJCAI*. AAAI Press, 2230–2236.

[48] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *SIGIR*. 331–338.

[49] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science*. 214–226.

[50] Harrison Edwards and Amos J. Storkey. 2016. Censoring Representations with an Adversary. In *ICLR*.

[51] Michael D. Ekstrand, Mucun Tian, Mohammed R. Imran Kazi, Hoda Mehrpouyan, and Daniel Kluver. 2018. Exploring author gender in book rating and recommendation. In *RecSys*. 242–250.

[52] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention harvesting for context-dependent examination-bias estimation. In *SIGIR*. 825–834.

[53] Golnoosh Farnadi, Pigi Kouki, Spencer K. Thompson, Sriram Srinivasan, and Lise Getoor. 2018. A Fairness-aware Hybrid Recommender System. *CoRR* abs/1809.09030 (2018).

[54] Batya Friedman and Helen Nissenbaum. 1996. Bias in Computer Systems. *ACM Trans. Inf. Syst.* 14, 3 (1996), 330–347.

[55] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, et al. 2021. Towards Long-term Fairness in Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 445–453.

[56] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. 2020. Understanding Echo Chambers in E-commerce Recommender Systems. In *SIGIR*. 2261–2270.

[57] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *KDD*. 2221–2231.

[58] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. 2016. The case for process fairness in learning: Feature selection for fair decision making. In *NeurIPS Symposium on Machine Learning and the Law*, Vol. 1. 2.

[59] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. 2009. Click chain model in web search. In *WWW*. 11–20.

[60] Ruocheng Guo, Xiaoting Zhao, Adam Henderson, Liangjie Hong, and Huan Liu. 2020. Debiasing Grid-based Product Search in E-commerce. In *KDD*. 2852–2860.

[61] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *NeurIPS*. 3315–3323.

[62] David Haussler. 1990. *Probably approximately correct learning*. University of California, Santa Cruz, Computer Research Laboratory.

[63] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. 639–648.

[64] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. ACM, 173–182.

[65] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR*. ACM, 549–558.

[66] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-random Missing Data.. In *ICML*. 1512–1520.

[67] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten De Rijke. 2013. Reusing historical interaction data for faster online learning to rank for IR. In *WSDM*. 183–192.

[68] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*. Ieee, 263–272.

[69] Rashidul Islam, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds. 2021. Debiasing career recommendations with neural fair collaborative filtering. In *Proceedings of the Web Conference 2021*. 3779–3790.

[70] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *WSDM*. 447–455.

[71] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction* 25, 5 (2015), 427–491.

[72] Olivier Jeunen, David Rohde, Flavian Vasile, and Martin Bompaire. 2020. Joint Policy-Value Learning for Recommendation. In *KDD*. 1223–1233.

[73] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. 2019. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 383–390.

[74] Jiarui Jin, Yuchen Fang, Weinan Zhang, Kan Ren, Guorui Zhou, Jian Xu, Yong Yu, Jun Wang, Xiaoqiang Zhu, and Kun Gai. 2020. A Deep Recurrent Survival Model for Unbiased Ranking. *arXiv preprint arXiv:2004.14714* (2020).

[75] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, Vol. 51. Acm New York, NY, USA, 4–11.

[76] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)* 25, 2 (2007), 7–es.

[77] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *WSDM*. 781–789.

[78] Christopher C Johnson. 2014. Logistic matrix factorization for implicit feedback data. *NeurIPS* 27 (2014).

[79] Matthew Joseph, Michael J. Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. 2016. Rawlsian Fairness for Machine Learning. *CoRR* abs/1610.09559 (2016).

[80] Toshihiro Kamishima and Shotaro Akaho. 2017. Considerations on Recommendation Independence for a Find-Good-Items Task. In *FATREC Workshop on Responsible Recommendation Proceedings*.

[81] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Enhancement of the Neutrality in Recommendation. In *RecSys*, Vol. 893. 8–14.

[82] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2013. Efficiency Improvement of Neutrality-Enhanced Recommendation.. In *Decisions@ RecSys*. 1–8.

[83] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2014. Correcting Popularity Bias by Enhancing Recommendation Neutrality. In *RecSys*.

[84] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Issei Sato. 2016. Model-Based Approaches for Independence-Enhanced Recommendation. In *ICDM*. 860–867.

[85] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. 2018. Homophily influences ranking of minorities in social networks. *Scientific reports* 8, 1 (2018), 1–12.

[86] Joon Sik Kim, Jiahao Chen, and Ameet Talwalkar. 2020. FACT: A Diagnostic for Group Fairness Trade-offs. *ICML* (2020).

[87] Yong-Deok Kim and Seungjin Choi. 2014. Bayesian binomial mixture model for collaborative prediction with non-random missing data. In *Proceedings of the 8th ACM Conference on Recommender systems*. 201–208.

[88] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *ITCS (LIPIcs)*, Vol. 67. 43:1–43:23.

[89] Kerstin Klöckner, Nadine Wirschum, and Anthony Jameson. 2004. Depth-and breadth-first processing of search result lists. In *CHI'04 extended abstracts on Human factors in computing systems*. 1539–1539.

[90] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[91] Adit Krishnan, Ashish Sharma, Aravind Sankar, and Hari Sundaram. 2018. An Adversarial Approach to Improve Long-Tail Performance in Neural Collaborative Filtering. In *CIKM*. 1491–1494.

[92] Sanjay Krishnan, Jay Patel, Michael J Franklin, and Ken Goldberg. 2014. A methodology for learning, analyzing, and mitigating social influence bias in recommender systems. In *RecSys*. 137–144.

[93] Matt J. Kusner, Joshua R. Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual Fairness. In *NeurIPS*. 4066–4076.

[94] Anja Lambrecht and Catherine Tucker. 2019. Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads. *Management Science* 65, 7 (2019), 2966–2981.

[95] Gael Lederrey and Robert West. 2018. When sheep shop: measuring herding effects in product ratings with natural experiments. In *Proceedings of the 2018 World Wide Web Conference*. 793–802.

[96] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*. 661–670.

[97] Roger Zhe Li, Julián Urbano, and Alan Hanjalic. 2021. Leave No User Behind: Towards Improving the Utility of Recommender Systems for Non-mainstream Users. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 103–111.

[98] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2021. User-oriented Fairness in Recommendation. In *Proceedings of the Web Conference 2021*. 624–632.

[99] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards Personalized Fairness based on Causal Notion. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 1054–1063.

[100] Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. 2010. Improving one-class collaborative filtering by incorporating rich user information. In *CIKM*. 959–968.

[101] Defu Lian, Rui Liu, Yong Ge, Kai Zheng, Xing Xie, and Longbing Cao. 2017. Discrete content-aware matrix factorization. In *KDD*. ACM, 325–334.

[102] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *WWW*. ACM, 951–961.

[103] Jianghao Lin, Weiwen Liu, Xinyi Dai, Weinan Zhang, Shuai Li, Ruiming Tang, Xiuqiang He, Jianye Hao, and Yong Yu. 2021. A Graph-Enhanced Click Model for Web Search. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1259–1268.

[104] Kun Lin, Nasim Sonboli, Bamshad Mobasher, and Robin Burke. 2019. Crank up the Volume: Preference Bias Amplification in Collaborative Recommendation. In *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, September 20, 2019*, Vol. 2440. CEUR-WS.org.

[105] Xiao Lin, Min Zhang, Yongfeng Zhang, Zhaoquan Gu, Yiqun Liu, and Shaoping Ma. 2017. Fairness-Aware Group Recommendation with Pareto-Efficiency. In *RecSys*. 107–115.

[106] Zinan Lin, Dugang Liu, Weike Pan, and Zhong Ming. 2021. Transfer Learning in Collaborative Recommendation for Bias Reduction. In *Fifteenth ACM Conference on Recommender Systems*. 736–740.

[107] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data. In *SIGIR*. 831–840.

[108] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. Mitigating Confounding Bias in Recommendation via Information Bottleneck. In *Fifteenth ACM Conference on Recommender Systems*. 351–360.

[109] Weiwen Liu, Jun Guo, Nasim Sonboli, Robin Burke, and Shengyu Zhang. 2019. Personalized fairness-aware re-ranking for microlending. In *RecSys*. 467–471.

[110] Yiming Liu, Xuezhi Cao, and Yong Yu. 2016. Are You Influenced by Others When Rating?: Improve Rating Prediction by Conformity Modeling. In *RecSys*. ACM, 269–272.

[111] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard S. Zemel. 2016. The Variational Fair Autoencoder. In *ICLR*.

[112] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *SIGIR*. ACM, 203–210.

[113] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *SIGIR*. 1137–1140.

[114] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. Feedback Loop and Bias Amplification in Recommender Systems. *arXiv preprint arXiv:2007.13019* (2020).

[115] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *RecSys*. 5–12.

[116] Benjamin M Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative filtering and the missing at random assumption. In *UAI*. 267–275.

[117] James McInerney, Brian Brost, Praveen Chandar, Rishabh Mehrotra, and Benjamin Carterette. 2020. Counterfactual Evaluation of Slate Recommendations with Sequential Reward Interactions. In *KDD*. 1779–1788.

[118] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* (2019).

[119] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A Survey on Bias and Fairness in Machine Learning. *CoRR* abs/1908.09635 (2019).

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

[121] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. In *SIGIR*. ACM, 429–438.

[122] Razieh Nabi and Ilya Shpitser. 2018. Fair Inference on Outcomes. In *AAAI*. 1931–1940.

[123] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasanakis, et al. 2020. Bias in data-driven artificial intelligence systems—An introductory survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10, 3 (2020), e1356.

[124] Harrie Oosterhuis. 2021. Computationally Efficient Optimization of Plackett-Luce Ranking Models for Relevance and Fairness. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 1023–1032.

[125] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for Selection Bias in Learning-to-rank Systems. In *Proceedings of The Web Conference 2020*. 1863–1873.

[126] Zohreh Ovaisi, Kathryn Vasilaky, and Elena Zheleva. 2021. Propensity-Independent Bias Recovery in Offline Learning-to-Rank Systems. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 1763–1767.

[127] Maeve O'Brien and Mark T Keane. 2006. Modeling result-list searching in the World Wide Web: The role of relevance topologies and trust bias. In *Proceedings of the 28th annual conference of the cognitive science society*, Vol. 28. Citeseer, 1881–1886.

[128] Rong Pan and Martin Scholz. 2009. Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In *KDD*. 667–676.

[129] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. *ICDM* (2008), 502–511. https://doi.org/10.1109/ICDM.2008.16

[130] Dae Hoon Park and Yi Chang. 2019. Adversarial Sampling and Training for Semi-Supervised Information Retrieval. In *The World Wide Web Conference*. ACM, 1443–1453.

[131] Judea Pearl. 2009. *Causality*. Cambridge university press.

[132] Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

[133] Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *KDD*. 560–568.

[134] Zhen Qin, Suming J Chen, Donald Metzler, Yongwoo Noh, Jingzheng Qin, and Xuanhui Wang. 2020. Attribute-based Propensity for Unbiased Learning in Recommender Systems: Algorithm and Case Studies. In *KDD*. 2359–2367.

[135] Tahleen A. Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *IJCAI*. 3289–3295.

[136] Karthik Raman and Thorsten Joachims. 2013. Learning socially optimal information systems from egoistic users. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 128–144.

[137] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*. ACM, 273–282.

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

[139] Nir Rosenfeld, Yishay Mansour, and Elad Yom-Tov. 2017. Predicting counterfactuals from large historical data and small randomized trials. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 602–609.

[140] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-Not-At-Random Explicit Feedback. In *SIGIR*. 309–318.

[141] Yuta Saito. 2020. Unbiased Pairwise Learning from Biased Implicit Feedback. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 5–12.

[142] Tobias Schnabel and Paul N Bennett. 2020. Debiasing item-to-item recommendations with small annotated datasets. In *Fourteenth ACM Conference on Recommender Systems*. 73–81.

[143] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML (JMLR Workshop and Conference Proceedings)*, Vol. 48. 1670–1679.

[144] Anne Schuth, Harrie Oosterhuis, Shimon Whiteson, and Maarten de Rijke. 2016. Multileave gradient descent for fast online learning to rank. In *WSDM*. 457–466.

[145] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624* (2021).

[146] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *KDD*. 2219–2228.

[147] Ashudeep Singh and Thorsten Joachims. 2019. Policy Learning for Fairness in Ranking. In *Conference on Neural Information Processing Systems (NeurIPS 2019)*.

[148] Ayan Sinha, David F Gleich, and Karthik Ramani. 2016. Deconvolving feedback loops in recommender systems. In *Advances in neural information processing systems*. 3243–3251.

[149] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *KDD*. 713–722.

[150] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *RecSys*. 213–220.

[151] Ana-Andreea Stoica, Christopher J. Riederer, and Augustin Chaintreau. 2018. Algorithmic Glass Ceiling in Social Networks: The effects of social recommendations on network diversity. In *WWW*. 923–932.

[152] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the human-recommender system feedback loop in collaborative filtering. In *Companion Proceedings of The 2019 World Wide Web Conference*. 645–651.

[153] Adith Swaminathan and Thorsten Joachims. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research* 16, 1 (2015), 1731–1755.

[154] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*. 3632–3642.

[155] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: discerning multi-faceted trust in a connected world. In *WSDM*. ACM, 93–102.

[156] John K Tarus, Zhendong Niu, and Ghulam Mustafa. 2018. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial intelligence review* 50, 1 (2018), 21–48.

[157] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade Model-based Propensity Estimation for Counterfactual Learning to Rank. In *SIGIR*. 2089–2092.

[158] Ali Vardasbi, Harrie Oosterhuis, and Maarten de Rijke. 2020. When Inverse Propensity Scoring does not Work: Affine Corrections for Unbiased Learning to Rank. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1475–1484.

[159] Francis Vella. 1998. Estimating models with sample selection bias: a survey. *Journal of Human Resources* (1998), 127–169.

[160] Can Wang, Jiawei Chen, Sheng Zhou, Qihao Shi, Yan Feng, and Chun Chen. 2021. SamWalker++: recommendation with informative sampling strategy. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[161] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization bandits for interactive recommendation.. In *AAAI*, Vol. 17. 2695–2702.

[162] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *SIGIR*. ACM, 515–524.

[163] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. 2018. Collaborative Filtering With Social Exposure: A Modular Approach to Social Recommendation. In *AAAI*.

[164] Ningxia Wang and Li Chen. 2021. User Bias in Beyond-Accuracy Measurement of Recommendation Algorithms. In *Fifteenth ACM Conference on Recommender Systems*. 133–142.

[165] Nan Wang, Zhen Qin, Xuanhui Wang, and Hongning Wang. 2021. Non-Clicks Mean Irrelevant? Propensity Ratio Scoring As a Correction. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 481–489.

[166] Ting Wang and Dashun Wang. 2014. Why Amazon's ratings might mislead you: The story of herding effects. *Big data* 2, 4 (2014), 196–204.

[167] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 1717–1725.

[168] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2020. " Click" Is Not Equal to" Like": Counterfactual Recommendation for Mitigating Clickbait Issue. *arXiv preprint arXiv:2009.09945* (2020).

[169] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *SIGIR*. 115–124.

[170] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. 2018. A reinforcement learning framework for explainable recommendation. In *ICDM*. IEEE, 587–596.

[171] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *WSDM*. 610–618.
[172] Xin Wang, Steven CH Hoi, Martin Ester, Jiajun Bu, and Chun Chen. 2017. Learning personalized preference of strong and weak ties for social recommendation. In *WWW*. IW3C2, 1601–1610.
[173] Xin Wang, Steven CH Hoi, Chenghao Liu, and Martin Ester. 2017. Interactive social recommendation. In *CIKM*. 357–366.
[174] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *www*. ACM / IW3C2, 99–109.
[175] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *Proceedings of The Web Conference 2020*. 99–109.
[176] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *ICML*. 6638–6647.
[177] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2021. Combating Selection Biases in Recommender Systems with a Few Unbiased Ratings. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 427–435.
[178] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2018. The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581* (2018).
[179] Jacek Wasilewski and Neil Hurley. 2016. Incorporating Diversity in a Learning to Rank Recommender System. In *FLAIRS*. 572–578.
[180] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1791–1800.
[181] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire Space Multi-Task Modeling via Post-Click Behavior Decomposition for Conversion Rate Prediction. In *SIGIR*. 2377–2386.
[182] Robert West, Smriti Bhagat, Paul Groth, Marinka Zitnik, Francisco M Couto, Pasquale Lisena, Albert Meroño-Peñuela, Xiangyu Zhao, Wenqi Fan, Dawei Yin, et al. 2021. Summary of Tutorials at The Web Conference 2021. In *Companion Proceedings of the Web Conference 2021*. 727–733.
[183] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. 2021. Learning Fair Representations for Recommendation: A Graph-based Perspective. In *Proceedings of the Web Conference 2021*. 2198–2208.
[184] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM*. ACM, 153–162.
[185] Yongkai Wu, Lu Zhang, and Xintao Wu. 2018. On Discrimination Discovery and Removal in Ranked Data using Causal Graph. In *KDD*. ACM, 2536–2544.
[186] Yongkai Wu, Lu Zhang, and Xintao Wu. 2019. Counterfactual Fairness: Unidentification, Bound and Algorithm.. In *IJCAI*. 1438–1444.
[187] Shuyuan Xu, Juntao Tan, Shelby Heinecke, Jia Li, and Yongfeng Zhang. 2021. Deconfounded Causal Collaborative Filtering. *arXiv preprint arXiv:2110.07122* (2021).
[188] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. 2021. Policy-Gradient Training of Fair and Unbiased Ranking Functions. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1044–1053.
[189] Haiqin Yang, Guang Ling, Yuxin Su, Michael R Lyu, and Irwin King. 2015. Boosting response aware model-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering* 27, 8 (2015), 2064–2077.
[190] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*. 279–287.
[191] Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. 2021. Top-N Recommendation with Counterfactual User Preference Simulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2342–2351.
[192] Sirui Yao and Bert Huang. 2017. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *NeurIPS*. 2921–2930.
[193] Sirui Yao and Bert Huang. 2017. New Fairness Metrics for Recommendation that Embrace Differences. *CoRR* abs/1706.09838 (2017).
[194] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin. 2017. Selection of negative samples for one-class matrix factorization. In *SDM*. SIAM, 363–371.
[195] Jiangxing Yu, Hong Zhu, Chih-Yao Chang, Xinhua Feng, Bowen Yuan, Xiuqiang He, and Zhenhua Dong. 2020. Influence Function for Unbiased Recommendation. In *SIGIR*. 1929–1932.
[196] Bowen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. 2019. Improving ad click prediction by considering non-displayed events. In *CIKM*. 329–338.
[197] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *WWW*. 1171–1180.
[198] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. Fairness Constraints: Mechanisms for Fair Classification. In *AISTATS*, Vol. 54. 962–970.

[199] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A Fair Top-k Ranking Algorithm. In *CIKM*. 1569–1578.

[200] Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. 2013. Learning Fair Representations. In *ICML*, Vol. 28. 325–333.

[201] Hanwang Zhang, Fumin Shen, Wei Liu, Xiangnan He, Huanbo Luan, and Tat-Seng Chua. 2016. Discrete collaborative filtering. In *SIGIR*. ACM, 325–334.

[202] Junzhe Zhang and Elias Bareinboim. 2018. Fairness in Decision-Making - The Causal Explanation Formula. In *AAAI*. 2037–2045.

[203] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[204] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-scale Causal Approaches to Debiasing Post-click Conversion Rate Estimation with Multi-task Learning. In *Proceedings of The Web Conference 2020*. 2775–2781.

[205] Wei Vivian Zhang and Rosie Jones. 2007. Comparing click logs and editorial labels for training query rewriting. In *WWW 2007 Workshop on Query Log Analysis: Social And Technological Challenges*.

[206] Xueru Zhang and Mingyan Liu. 2020. Fairness in Learning-Based Sequential Decision Algorithms: A Survey. *arXiv preprint arXiv:2001.04861* (2020).

[207] Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Found. Trends Inf. Retr.* 14, 1 (2020), 1–101.

[208] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 11–20.

[209] Xiangyu Zhao, Long Xia, Jiliang Tang, and Dawei Yin. 2019. Deep reinforcement learning for search, recommendation, and online advertising: a survey. *ACM SIGWEB Newsletter* Spring (2019), 1–15.

[210] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *RecSys*. 95–103.

[211] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *KDD*. 1040–1048.

[212] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2017. Deep reinforcement learning for list-wise recommendations. *arXiv preprint arXiv:1801.00209* (2017).

[213] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *CIKM*. 1411–1420.

[214] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. 2020. Jointly Learning to Recommend and Advertise. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. 3319–3327.

[215] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2021. Popularity Bias Is Not Always Evil: Disentangling Benign and Harmful Bias for Recommendation. *arXiv preprint arXiv:2109.07946* (2021).

[216] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. 167–176.

[217] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.

[218] Ke Zhou and Hongyuan Zha. 2012. Learning binary codes for collaborative filtering. In *KDD*. ACM, 498–506.

[219] Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. 2020. Unbiased Implicit Recommendation and Propensity Estimation via Combinational Joint Learning. In *Fourteenth ACM Conference on Recommender Systems*. 551–556.

[220] Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity Bias in Dynamic Recommendation. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 2439–2449.

[221] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-Opportunity Bias in Collaborative Filtering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 85–93.

[222] Ziwei Zhu, Xia Hu, and James Caverlee. 2018. Fairness-Aware Tensor-Based Recommendation. In *CIKM*. 1153–1162.

[223] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and mitigating item under-recommendation bias in personalized ranking systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 449–458.

[224] Zeyuan Allen Zhu, Weizhu Chen, Tom Minka, Chenguang Zhu, and Zheng Chen. 2010. A novel click model and its applications to online advertising. In *WSDM*. 321–330.