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

Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, Xiangnan He

Abstract—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.

Index Terms—Recommendation, Recommender System, Collaborative Filtering, Survey, Bias, Debias, Fairness

✦

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 (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 2018 and 2020 to discuss topics on bias elimination[1]. SIGIR even presents the Best Paper award to the paper on this topic in 2018 [1] and 2020 [2], respectively. 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

- *Jiawei Chen, Hande Dong, Xiangnan He are with the University of Science and Technology of China. E-mail: cjwustc@ustc.edu.cn, donghd@mail.ustc.edu.cn and hexn@ustc.edu.cn.*

- *Xiang Wang, Fuli Feng are with the National University of Singapore. E-mail: xiangwang@u.nus.edu and fulifeng93@gmail.com.*

- *Meng Wang is with the Hefei University of Technology. E-mail: eric.mengwang@gmail.com.*

- *Xiangnan He is corresponding author*

1. http://www.sigir.org/sigir2020/schedule/; http://sigir.org/sigir2018/program/program-at-a-glance/
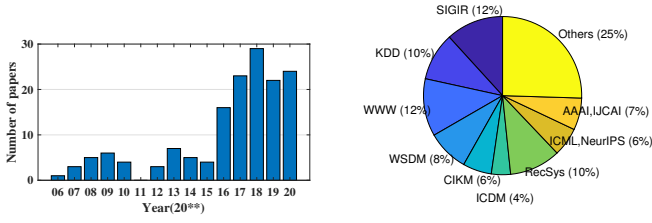2. https://tianchi.aliyun.com/competition/entrance/231785/introduction

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

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 [3], while others use "observational bias" to refer to the same meaning instead [4]. More confusingly, the same terminology "selection bias" has been conceptualized differently in different publications [3], [5], [6]. Moreover, a considerable number of researchers do not explicitly mention "bias" or "debias" in the paper (e.g. [7], [8], [9]), 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, [10] reviews explainable recommendation, [11] reviews knowledge-based recommendation, [12] and [13] 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, [14] recently reviews the bias issues in natural language processing, [15] reviews the sample selection bias on model estimation [15], and [16] summarizes fairness in learning-based sequential decision algorithms. There are some surveys on the bias and fairness of general machine learning and artificial intelligence systems [17], [18], [19]. 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.
- Conducting a comprehensive review and providing a taxonomy of existing methods on recommendation debiasing.
- Identifying open challenges and discussing future directions to inspire more research on this topic.

Papers Collection. We collect over 140 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, inculding 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 2020. 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 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**, 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). We denote the collected interactions between the user set $\mathcal{U} = \{u_1, \cdots, u_n\}$ and item set $\mathcal{I} = \{i_1, \cdots, i_m\}$, as a feedback matrix. In general, there are two types of user feedback: 1) implicit feedback, which is represented as a matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, where each entry x_{ui} in $\mathbf{X}$ is binarized to indicate whether the user u has interacted with the item i (*e.g.*, purchase, click, view) before or not; and 2) explicit feedback, which is represented as a matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$, where each entry r_{ui} in $\mathbf{R}$ is real-valued that directly reflects user preference on the rated items.
- **Data→Model**, 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**, 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
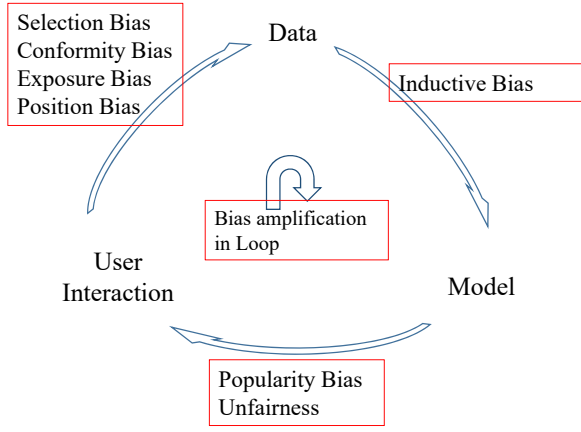
Fig. 2. Feedback loop in recommendation, where biases occur in different stages.

(a) Random   (b) User-selected

Fig. 3. Distribution of rating values for randomly selected items and user-selected items. The data is from [20] with permission.

the RS can lead to a self-reinforcing pattern by leveraging the updated data.

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 these biases in terms of bias amplification.

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. Here we categorize the data biases into four groups: selection bias and conformity bias in explicit feedback, exposure bias and position bias in implicit feedback.

3.1.1 Bias in explicit feedback data

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.* [20] 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 3 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 [4], [21].
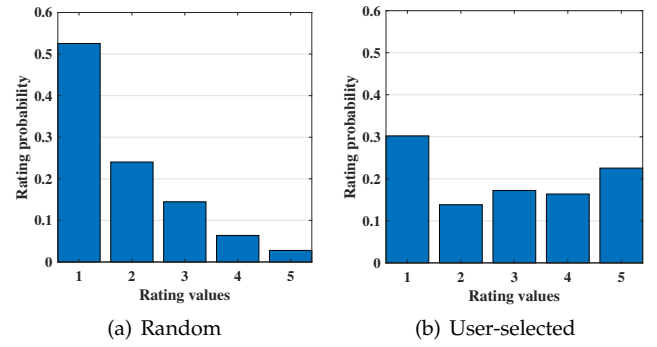
Another bias inherent in the explicit feedback data is conformity bias, which is defined as:

- **Conformity Bias.** *Conformity bias happens as users tend to rate similarly to the others in a group, even if doing so goes against their own judgment, making the rating values 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 [22], [23]. Such phenomenon of conformity is common and cause biases in user ratings. As shown in Krishnan *et al.* [24], 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 [25], [26], [27], [28]. Hence, the observed ratings are skewed and might not reflect users' real preference on items [23].

3.1.2 Bias in implicit feedback data

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. Hence, some biases originate from such one-class data such as exposure bias and position bias.

Exposure bias occurs in real-world RS with massive items, 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 [29]. Hence, some recent works [29] also name such "exposure bias" as "previous mode bias". 2) As users may actively search and find

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

Types	Stages in Loop	Data	Cause	Effect	Major solutions
Selection Bias	User→Data	Explicit feedback	Users' self-selection	Skewed observed rating distribution	Data imputation; Propensity Score; Doubly robust model
Conformity Bias	User→Data	Explicit feedback	Conformity	Skewed rating values	Modeling social or popularity effect
Exposure Bias	User→Data	Implicit feedback	Users' self-selection; Intervened by systems	Unreliable non-positive data	Give confidence weights by heuristic, sampling or exposure-based model
Position Bias	User→Data	Implicit feedback	Trust top of lists; Exposed to top of lists	Unreliable positive data	Click models; Propensity Score
Inductive Bias	Data→Model	Both	Added by researchers	Better generalization, lower variance or faster recommendation	-
Popularity Bias	Model→User	Both	Algorithm and unbalanced data	Matthew effect	Regularization; Adversarial learning; Causal graph
Unfairness	Model→User	Both	Algorithm and unbalanced data	Unfairness for some groups	Rebalancing; Regularization; Adversarial learning; Causal modeling
Bias amplification in Loop	All	Both	Feedback loop	Enhance and spread bias	Break the loop by collecting random data or using reinforcement learning

the items of interest, the selection of users is a factor of exposure [5], [6], and makes highly relevant items more likely to be exposed. Hence, in this scenario, "exposure bias" is named as "selection bias". 3) The background of users is another factor to expose items, such as social friends [30], 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" [32]. 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.

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.*

It 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 [33]. Here "relevance" is widely used in the field of information retrieval, which denotes how the items are preferred by the users. 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 [34], [35]. Moreover, Maeve *et al.* [36] shows that users often trust the first few results in the lists and then stop assessing the rest, without evaluating the entire list holistically [37]. As such, the data collected from user feedback towards the recommended lists may fail to reflect user preference faithfully [33]. Therefore, position bias poses challenges in both training and evaluation phases.

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.* [38] assumes an interaction can be estimated by embedding inner product, while He *et al.* [39] 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 [40], [41], [42], [43], 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 [44], [45], [46] which embeds user and items as binary codes to improve the efficiency of recommendation, which is at the expense of sacrificing the representation ability.

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 [47].*

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 [47]. 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.* [47]. Figure 4 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 [47].

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
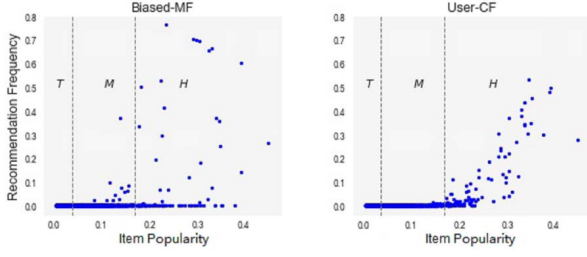
Fig. 4. Item popularity VS. recommendation frequency (Biased-MF [48] and User-CF [49]), where items were classified into three different groups: H denoting the set of most popular items that take up around 20% entire ratings, T denoting the set of most unpopular items that take up 20% entire ratings and M denotes the rest. The figure was reproduced from [47] with authors' permission.

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. 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.

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*" [50], 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 [51].*

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). For example, in the context of job recommendation, previous work [52], [53] found that, compared to men, women saw less ads about high paying jobs and career coaching services, which is 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 [54], [55]. Another similar issue has been found in book recommendation, where the methods prefer recommending books of male authors [56]. 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. 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 [36]. Many researchers also study the impact of feedback loop on the popularity bias [57], [58], [59]. 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" [60], [61].

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. we classify them according to which biases they addressed and which types of methods they adopted.

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 [62]:

$$H(\hat{R}) = \frac{1}{nm} \sum_{u=1}^{n} \sum_{i=1}^{m} \delta_{u,i}(r, \hat{r}) \tag{1}$$

for an appropriately chosen $\delta_{u,i}(r, \hat{r})$:

$$\text{MAE: } \delta_{u,i}(r, \hat{r}) = |r_{u,i} - \hat{r}_{u,i}| \tag{2}$$

$$\text{MSE: } \delta_{u,i}(r, \hat{r}) = (r_{u,i} - \hat{r}_{u,i})^2 \tag{3}$$

$$\text{DCG@k} : \delta_{u,i}(r, \hat{r}) = (I / \log(\text{rank}(\hat{r}_{u,i}))) \, r_{u,i} \tag{4}$$

$$\text{Pre@k} : \delta_{u,i}(r, \hat{r}) = (I/k) r_{u,i} \cdot \mathbf{1} \{\text{rank}(\hat{r}_{u,i}) \leq k\} \tag{5}$$

where $r_{u,i}$ denotes the true rating values of the item i given by the user u and $\hat{r}_{u,i}$ denotes the predicted rating values by the recommendation model. As true ratings r are usually partially observed, the conventional evaluation usually use the average over only the observed entries:

$$\hat{H}_{\text{naive}}(\hat{r}) = \frac{1}{|\{(u,i) : O_{u,i} = 1\}|} \sum_{(u,i):O_{u,i}=1} \delta_{u,i}(r, \hat{r}) \tag{6}$$

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

Addressed issues	Categories		Publications
Selection Bias	Evaluator	Propensity Score	[62]
		ATOP	[63]
	Training	Data imputation	[4], [20], [21], [63], [64], [65], [66]
		Propensity Score	[62]
		Doubly robust model	[67]
		Meta learning	[3]
Conformity Bias	Modeling popularity influence		[23]
	Modeling social influence		[25], [26], [27], [28]
Exposure Bias	Evaluator	Propensity Score	[68]
	Training	Heuristic	[69], [70], [71], [72], [73], [74], [75], [76]
		Sampling	[30], [43], [74], [77], [78], [79]
		Exposure-based model	[7], [8], [30], [31], [80]
		Others	[5], [6], [81], [82], [83], [84], [85]
Position Bias	Click models		[86], [87], [88], [89], [90], [91], [92]
	Propensity Score		[5], [93], [94], [94], [94], [95], [96], [97], [98], [99], [100], [101], [102], [103], [104], [105]
Popularity Bias	Regularization		[106], [107], [108], [109]
	Adversarial learning		[110]
	Causal graph		[32]
	Others		[111], [112]
Unfairness	Rebalancing		[113], [114], [114], [115], [116], [117], [118], [119], [120], [121]
	Regularization		[106], [122], [123], [124], [125], [126], [127], [128], [129], [130], [131]
	Adversarial learning		[132], [133], [134]
	Causal modeling		[135], [136], [137], [137], [138], [138], [139]
Loop effect	Uniform data		[29], [29], [60], [140], [141], [142], [143]
	Reinforcement learning		[13], [144], [145], [146], [147], [148], [149], [150], [151], [152], [153], [154], [155], [156], [156], [157], [158], [159], [160]
	Others		[161], [162]

which is not an unbiased estimate of the true performance [62]:

$$E_O\left[\hat{H}_{\text{naive}}(\hat{r})\right] \neq H(\hat{r}) \tag{7}$$

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 [62] 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 $P_{u,i}$, which is defined as the marginal probability of observing a rating value ($P_{u,i} = P(O_{u,i} = 1)$) for certain user-item pair (u, i), can offset the selection bias. The proposed estimator is defined as:

$$\hat{H}_{IPS}(\hat{r} \mid P) = \frac{1}{nm} \sum_{(u,i):O_{u,i}=1} \frac{\delta_{u,i}(r,\hat{r})}{P_{u,i}} \tag{8}$$

which is an unbiased estimator of the ideal metric:

$$\mathbb{E}_O\left[\hat{H}_{IPS}(\hat{r} \mid P)\right] = \frac{1}{nm} \sum_u \sum_i \mathbb{E}_{O_{u,i}}\left[\frac{\delta_{u,i}(r,\hat{r})}{P_{u,i}} O_{u,i}\right]$$
$$= \frac{1}{nm} \sum_u \sum_i \delta_{u,i}(r,\hat{r}) = H(\hat{r}) \tag{9}$$

ATOP. Steck *et al.* [63] 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:

$$\text{TOPK}_u^{\text{obs}}(k) = \frac{N_u^{+,\text{obs},k}}{N_u^{+,\text{obs}}} \tag{10}$$

$$\text{TOPK}^{\text{obs}}(k) = \sum_u w^u \text{TOPK}_u^{\text{obs}}(k) \tag{11}$$

$$\text{ATOP}^{\text{obs}} = \int_0^1 \text{TOPK}^{\text{obs}}(k)dk \tag{12}$$

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 [3]. 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

Data imputation. Note that the main reason for the selection bias in the observed rating data 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 (i.e. 'which rating value the user gives') and missing data prediction task (i.e. 'which items the user select to rate'). For example, some recent work [4], [20], [64], [65] treat the problem as missing data imputation and jointly model the generative process of rating values and the missing mechanism. The basic assumption behind these methods is that the probability of users' selection on items depends on users' rating values for that item. Correspondingly, the missing probability of a user-item pair has been modeled dependent on the rating values in recent work with a Mixture of Multinomials [20], Logit model [64], MF model [4], [66] or social-enhanced model [65].

As joint learning the missing data model and the rating model will lead to sophisticated and highly complex methods, some light data imputation strategies have been proposed. Steck *et al.* [21], [63] directly impute the missing data with a specific value r_0 and optimize the model with the following objective function:

$$\arg\min_{\theta} \sum_{u,i} W_{u,i} \cdot \left(r_{u,i}^{o\&i} - \hat{r}_{u,i}(\theta) \right)^2 + Reg(\theta) \qquad (13)$$

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

However, as missing data model or imputed rating values are specified in a heuristic manner, this kind of methods will suffer from empirical inaccuracy due to mis-specified missing-data model or inaccurate imputed rating values. Such inaccuracy will be propagated into recommendation model training, resulting in sub-optimal recommendation performance [67].

Propensity score. Another kind of methods leverage inverse propensity score to mitigate the selection bias when training a recommendation model [62]. They directly use the IPS-based unbiased estimator as the objective and optimize the following empirical risk function:

$$\arg\min_{\theta} \sum_{O_{u,i}=1} \frac{\hat{\delta}_{u,i}\left(r, \hat{r}(\theta)\right)}{P_{u,i}} + Reg(\theta) \qquad (14)$$

where $\hat{\delta}_{u,i}\left(r, \hat{r}(\theta)\right)$ can be the original evaluation function or some surrogate objective functions for easy optimization; $Reg(\theta)$ denotes regularization term for mitigating over-fitting. Except for the propensities $P_{u,i}$ 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 it directly optimizes an unbiased empirical risk function.

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 [3], 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. [67] 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:

$$\mathcal{E}_{\mathrm{DR}} = \mathcal{E}_{\mathrm{DR}}\left(\hat{r}, r^o\right) = \frac{1}{nm} \sum_{u,i} \left(\hat{e}_{u,i} + \frac{o_{u,i} d_{u,i}}{\hat{p}_{u,i}} \right) \qquad (15)$$

where $\hat{e}_{u,i} = \delta_{u,i}(r_0, \hat{r}_{u,i})$ denotes imputed error between the predicted ratings and imputed ratings; $d_{u,i} = \hat{e}_{u,i} - \delta_{u,i}(r, \hat{r})$ denotes the error deviation between the imputed error and predicted error. The theoretical and empirical analyses presented in [67] 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.

Meta learning. To resolve the issue that the aforementioned methods highly rely on the choice of propensity and imputation, Saito *et al.* [3] propose a new meta-learning method with 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 [3] 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. The upper bound can be loose when A2 performs poorly, leading to sub-optimal results of A0. In fact, learning a satisfied A2 from biased data is challenging. A2 usually suffers from selection bias and the bias will be finally injected into the target recommendation model. Nevertheless, meta learning and upper bound optimization are promising directions for mitigating selection bias, which deserve future exploring.

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' ratings conform to public opinions. Correspondingly, Liu *et al.* [23] 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 [163]:

$$\hat{r}_{ui} = xgb\left(\left\{(1-\omega) \cdot t_{ui} + \omega \cdot a_{ui}, c_{ui}, a_{ui}, d_{ui}\right\}, \Theta_{xgb}\right) \qquad (16)$$

where t_{ui} denotes the prediction returned by basic recommendation model and ω controls the strength of conformity. This way, we can eliminate the effect caused by conformity and recover the user's true preference.

The other type of methods treat user's rating values as synthetic results of user preference and social influence [25], [26], [27], [28]. Thus, similar to [23], 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 propensity score, which also has been applied to address the selection bias. Yang *et al.* [68] 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:

$$R(\hat{Z}) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u|} \sum_{i \in \mathcal{S}_u} c\left(\hat{Z}_{u,i}\right) \tag{17}$$

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

$$\text{AUC}: c\left(\hat{Z}_{u,i}\right) = 1 - \frac{\hat{Z}_{u,i}}{|I|} \tag{18}$$

$$\text{DCG}: c\left(\hat{Z}_{u,i}\right) = \frac{1}{\log_2\left(\hat{Z}_{u,i} + 1\right)} \tag{19}$$

$$\text{DCG@k}: c\left(\hat{Z}_{u,i}\right) = \frac{1\left\{\hat{Z}_{u,i} \leq k\right\}}{\log_2\left(\hat{Z}_{u,i} + 1\right)} \tag{20}$$

$$\text{Recall@k}: c\left(\hat{Z}_{u,i}\right) = 1\left\{\hat{Z}_{u,i} \leq k\right\} \tag{21}$$

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

$$\hat{R}_{\text{AOA}}(\hat{Z}) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u^*|} \sum_{i \in S_u^*} c\left(\hat{Z}_{u,i}\right)$$
$$= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{\sum_{i \in S_u} O_{u,i}} \sum_{i \in S_u} c\left(\hat{Z}_{u,i}\right) \cdot O_{u,i} \tag{22}$$

where S_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.* [68] 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:

$$\hat{R}_{\text{IPS}}(\hat{Z} \mid P) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u|} \sum_{i \in S_u^*} \frac{c\left(\hat{Z}_{u,i}\right)}{P_{u,i}}$$
$$= \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|\mathcal{S}_u|} \sum_{i \in S_u} \frac{c\left(\hat{Z}_{u,i}\right)}{P_{u,i}} \cdot O_{u,i} \tag{23}$$

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

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 [3]:

$$\min_\theta \sum_{u,i} W_{ui} \delta\left(x_{ui}, \hat{r}_{ui}(\theta)\right) + Reg(\theta) \tag{24}$$

where x_{ui} denotes the interaction value for the user-item pair (u,i); $\delta(.)$ denotes the pre-defined loss function (e.g. cross-entropy loss) between the predicted preference $\hat{r}_{ui}(\theta)$ and feedback x_{ui}. W_{ui} denotes the confidence weight, which controls the confidence that the the feedback of user-item pair (u,i) should be predicted as x_{ui}. The specification of the confidence weight is critical to the model performance and can be roughly categorized into three types:

Heuristic. The first is heuristic-based strategy. For example, the classic weighted factorization matrix (WMF) [69] and dynamic MF [70] used a simple heuristic that the un-observed interactions are assigned with a uniform lower weight, i.e., $W_{ui} = 1$ for $x_{ui} = 1$ and $W_{ui} = c$ $(0 < c < 1)$ for $x_{ui} = 0$; Some researchers [71], [72] specify the confidence with based on user activity level, i.e., $W_{ui} = c_u, c_u \propto \sum_i x_{ui}$, as users associate with more items provide more reliable information; Analogously, item popularity has been considered to specify confidence weights [73], [74], as popular items are more probable to be exposed; Also, user-item feature similarity [75] has been considered to define the confidence.

Moreover, Saito *et al.* [76] leverage user's preference in evaluation and heuristically define an ideal loss function as follows:

$$\mathcal{L}_{\text{ideal}}(\theta) = \frac{1}{nm} \sum_{u,i} \left[r_{u,i}\delta\left(1, \hat{r}_{u,i}\right) + (1 - r_{u,i})\delta\left(0, \hat{r}_{u,i}\right)\right] \tag{25}$$

where each user-item interaction is weighted with the preference level r_{ui}. They further derive a surrogate estimator on implicit feedback with propensity scores as follows:

$$\hat{\mathcal{L}}_{\text{sur}}(\theta) = \frac{1}{nm} \sum_{u,i} \left[x_{ui}\left(\frac{1}{P_{ui}}\delta\left(1, \hat{r}_{u,i}\right) + \left(1 - \frac{1}{P_{ui}}\delta\left(0, \hat{r}_{u,i}\right)\right)\right.\right.$$
$$\left.\left. + (1 - x_{u,i})\delta\left(0, \hat{r}_{u,i}\right)\right] \tag{26}$$

---

3. We remark that here we just present the point-wise loss. In fact, pair-wise loss can be extended straightforwardly.

which is an unbiased estimator of the ideal loss function. Where the propensity score P_{ui} is defined as the marginal probability of a user exposed to the item.

However, assigning appropriate confidence weights heuristically is challenging, as the optimal data confidence may var for different user-item interactions. Choosing confidence weights or propensity scores usually require human experience or large computational resource for grid search. Furthermore, it is unrealistic to manually set personalized weights for millions of data. Coarse-grained manual confidence weights will cause 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:

$$E_{(u,i)\sim p}[\delta\left(x_{ui}, \hat{r}_{ui}(\theta)\right)] = \sum_{u,i} p_{ui}\delta\left(x_{ui}, \hat{r}_{ui}(\theta)\right) \quad (27)$$

where the sampling distribution acts as data confidence weights. Sampling strategy has been widely applied as its efficiency. For example, Logistical matrix factorization [38], BPR [77], or most of neural-based recommendation models (e.g. CDAE [164], NCF [39], LightGCN [165]) apply the uniform negative sampler; Yu et al. [74] 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. [43], [78] leverage viewed but non-clicked data to evaluate user's exposure; Chen et al. [30] leverage social network information in their sampling distribution; Wang et al. [79] 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 [8], [80]. EXMF [7] introduces an exposure variable and assumes the following generative process of implicit feedback:

$$a_{ui} \sim Bernoulli(\eta_{ui}) \quad (28)$$
$$(x_{ui}|a_{ui}=1) \sim Bernoulli(\hat{r}(\theta)) \quad (29)$$
$$(x_{ui}|a_{ui}=0) \sim \delta_0 \quad (30)$$

where a_{ui} denotes whether a user u has been exposed to the item i; δ_0 denotes delta function $p(x_{ui}=0|a_{ui}=0)=1$ and can be relaxed as Bernoulli distribution parameterized with a with small value; η_{ui} is the prior probability of exposure. When $a_{ui}=0$, we have $x_{ui} \approx 0$, since when the user does not know the item he can not interact with it. When $a_{ui}=1$, i.e., the user has known the item, he will decide whether or not to choose the item based on his preference. x_{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.
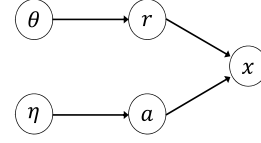

Fig. 5. Casual graph of exposure-based method.

Chen et al. [30], [31] give detailed analyses of EXMF and rewrite the objective function of EXMF as follows:

$$\sum_{ui} \gamma_{ui}\delta(x_{ui}, \hat{r}_{ui}(\theta)) + \sum_{ui} g(\gamma_{ui}) \quad (31)$$

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

$$g(\gamma_{ui}) = (1-\gamma_{ui})\ell(x_{ui}, \varepsilon) + \ell(\gamma_{ui}, \eta_{ui}) - \ell(\gamma_{ui}, \gamma_{ui}) \quad (32)$$

where $\ell(a,b) = a\log(b) + (1-a)\log(1-b)$. 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.

To better understand how exposure-based models remedy exposure bias, we conduct an analysis from the causal perspective. Let us draw an causal graph of exposure-based model as illustrated in Figure 5. The spirit of recommendation is to answer an intervention question: would the user purchase the items if he knows the item? That is, we need to evaluate the probability of $p(x_{ui}|do(a_{ui}=1), \theta, \eta)$. According to the back-door criteria [166], we have $p(x_{ui}|do(a_{ui}=1), \theta, \eta) = p(x_{ui}|a_{ui}=1, \theta, \eta) = p(\hat{r}(\theta))$. The performance of recommendation model indeed reflects the effect of recommendation.

However, directly estimating data confidence from Equation (31) 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 solution to re-parameterize the confidence weights with a simpler function. For example, Chen et al. [30] develop a social network-based function by modeling item information propagation along the social network; Chen et al. [31] consider that users are often exposed to information-sharing communities and thus replace confidence weights with a community-based function in their model.

Others. There are also some other strategies to address exposure bias in specific scenarios. Wang et al. [5] 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. [6] 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" and correspondingly devise an inverse propensity model [81], decomposition model [82], [83] or graph neural network [84] on the sequential behaviors to address exposure bias with multi-task learning. Besides, Wang *et al.* [85] consider the exposure bias from the perspective of item exposure features and leverages counterfactual reasoning to eliminate the direct causal effect from exposure features to the predicted preference.

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 in two lines.

4.4.1 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 [86], [87], [88], [89] 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 p as follows:

$$
\begin{aligned}
&P(C = 1 \mid u, i, p) \\
&= \underbrace{P(C = 1 \mid u, i, E = 1)}_{r_{ui}} \cdot \underbrace{P(E = 1 \mid p)}_{h_p}
\end{aligned}
\tag{33}
$$

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 [86]. 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_p, C_p be the probabilistic events indicating whether the p-th item is examined and clicked respectively. The cascade model generates users click data as follows:

$$
P(E_1) = 1 \tag{34}
$$
$$
P(E_{p+1} = 1 \mid E_p = 0) = 0 \tag{35}
$$
$$
P(E_{p+1} = 1 \mid E_p = 1, C_p) = 1 - C_p \tag{36}
$$
$$
P(C_p = 1 \mid E_p = 1) = r_{u_p,i} \tag{37}
$$

in which the Equation 36 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 $r_{u_p,i}$ and skipped with $1 - r_{u_p,i}$. This basic cascade model has been further improved by considering the personalized transition probability [88], [90], [91]. Jin *et al.* [92] 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 [5]. Further, mis-specifying the generative process of users clicks will cause empirical bias and hurt recommendation performance.

4.4.2 Propensity score

Another common solution to correct position bias is employing inverse propensity score, where each instance is weighted with a position-aware values [93]. The loss function is usually defined as follow:

$$
\begin{aligned}
L_{IPW}(S, q) &= \sum_{x \in \pi_q} \Delta_{IPW}(x, y \mid \pi_q) \\
&= \sum_{x \in \pi_q, o_q^x = 1, y = 1} \frac{\Delta(x, y \mid \pi_q)}{P(o_q^x = 1 \mid \pi_q)}
\end{aligned}
\tag{38}
$$

where Q denotes the universal set of all queries (or users) and q is an instance from Q. S denotes the ranking system and π_q is the ranked list retrieved by S for q, x be a document in π_q and y is the binary label that denotes whether x is relevant to q. Also, $\Delta(x, y \mid \pi_q)$ denotes a function that computes the individual loss on each relevant item. Joachims *et al.* [94] demonstrate the ranking model trained with the IPS-based loss function will converge to the same model trained with true relevance labels.

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 [5], [95], [96], [97], [98]. 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 [94] 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.* [99] and Agarwal *et al.* [100] adopt intervention harvesting, to learn the propensity. However, such methods require the feedback data from multiple ranking models. Further, some recent work [94], [101], [102], [104] consider learning a propensity model and a recommendation model as dual problem and develop specific EM algorithms to learn both models. Also, the click

model that captures the row skipping and slower decay phenomenon has been adopted to specify the propensity scores in [105], while cascade model has been adopted by [103].

4.5 Methods for Popularity Bias

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

4.5.1 Regularization

Suitable regularization can push the model towards balanced recommendation lists. Abdollahpouri *et al.* [106] introduced *LapDQ* regularizer [107] $tr(Q^T L_D Q)$, 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.* [108] applied the *mean-match* regularizer [167] 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 $M_D(\{\hat{r}\}) = \frac{1}{|D|} \sum_{(x_i,y_i,v_i) \in D} \hat{r}(x_i, y_i, v_i)$. Note that the above regularizers are result-oriented, guiding the model to give fairer results.

Different from result-oriented regularizers, Chen *et al.* [109] 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: $L_{DA} = \frac{1}{L^2} \sum_{(j,k)} (h_s^{jT} h_s^k - h_t^{jT} h_t^k)^2 = \frac{1}{L^2} ||Cov(D^s) - Cov(D^t)||_F^2$, 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 $Cov(D^s) = D^{sT}D^s$, and $Cov(D^t) = D^{tT}D^t$. 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.
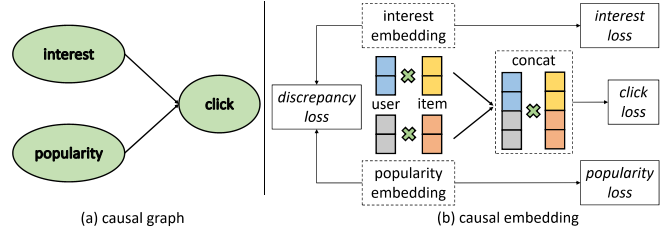

Fig. 6. Causal graph and causal embeddings [32]. Reproduced from [32] with authors' permission.

4.5.2 Adversarial learning

Adversarial learning is another line to address popularity bias*et al.* [110]. 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 [110], 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.

4.5.3 Causal graph

Causal graph is a powerful tool for counterfactual reasoning. Zheng *et al.* utilized causal inference to solve popularity bias [32]. They assumed user's click behavior on items depends on both interest and popularity, and construct a specific causal graph, as Figure 6 shows. To disentangle user interest and popularity bias, two kinds of embedding have been considered: interest embedding to capture users' real interest in items, and popularity embedding to capture pseudo-interest caused by popularity. these embeddings can be trained with cause-specific data under the framework of multi-task learning. Finally, interest embedding will be used for final recommendation, where popularity bias has been disentangled.

4.5.4 Others methods

There are some other methods on popularity bias. one solution to reduce popularity bias is thourgh introducing other side information. For example, Bressan *et al.* leverage social information to reduce popularity bias [111]. Abdollahpouri gives a different strategy [112], 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 [68].

4.6 Methods for Unfairness

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

4.6.1 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** [168]; 2) **individual fairness** [122], [169], [170], [171]; 3) **group fairness** (*e.g.,* demographic parity [172], [173], equality of opportunity [174], [175], predictive equality [176], equalized odds [174], calibration within groups [177]); and 4) **counterfactual fairness** [138]. 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 [178], spanning from fairness-aware ranking [114], [115], [116], supplier fairness in two-sided marketplace platforms [179], provider-side fairness to make items from different providers have a fair chance of being recommended [108], [180], fairness in group recommendation to minimize the unfairness between group members [129].

In the following sections, we review four different ways to mitigate the unfairness issue on recommendation.

4.6.2 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 [113], or re-sampling the training data to achieve statistical parity [114].

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 [115], [116] propose multiple ranking functions to sort items and then achieve fairness amortized across these rankings. Towards group fairness, FA∗IR [117] 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 [114] formalize the fairness as a desired distribution over sensitive attributes, and re-rank candidates (*i.e.,* LinkedIn users) to satisfy the constraints. To formulate group fairness in terms of exposure allocation, Singh *et al.* [118] propose a framework for formulating fairness constraints on rankings, and sample rankings from an associated probabilistic algorithm to fulfill the constraints. HyPER [119] 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 [120], [121] study potential unfairness issue inherent within graph embedding. Among them, Fairwalk [120] 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.

4.6.3 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 [122], 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:

$$\min \mathcal{L} = \alpha C(X, R) + \beta D(R, A) + \gamma E(Y, R) \quad (39)$$

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:

$$D(R, A) = |\mathbb{E}_R P(R|A = 1) - \mathbb{E}_R P(R|A = 0)| \quad (40)$$

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

$$P(R|A = 1) = \frac{\exp -||R - \tilde{R}_1||_2}{\sum_{a \in \{0,1\}} \exp -||R - \tilde{R}_a||_2} \quad (41)$$

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 [123], and then proposed two methods — (1) one regularization-based matrix completion method [124], 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 [125], 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 [126]. Analogously, Yao *et al.* [127], [128] 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.* [106] 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.* [129] worked on fairness-aware group recommendation, and designed a multi-objective optimization model to minimize the utility gap between group members. Burke *et al.* [130] 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.* [181] 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.* [131] 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. [2] 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:

$$\sigma_\tau = \underset{i \in \mathcal{I}}{\text{argsort}} \left(\hat{R}(i \mid u) + \lambda \operatorname{err}_\tau(i) \right) \quad (42)$$

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.

4.6.4 Adversarial Learning

Similar with the idea of LFR (*cf.* (39)), 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 Representation (ALFR)*, is proposed in [132] which is formulated as follows:

$$\max_\phi \min_\theta \mathcal{L} = \alpha C_\theta(X, R) + \beta D_{\theta,\phi}(R, A) + \gamma E_\theta(Y, R)$$
$$(43)$$

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:

$$D = \mathbb{E}_{X,A} A \cdot \log(f(R)) + (1 - A) \cdot \log(1 - f(R)) \quad (44)$$

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, Bose *et al.* [133] extended the ALFR framework by enforcing compositional fairness constraints on graph embeddings for multiple sensitive attributes, which are evaluated in the scenarios of movie and 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.* [134] 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.

4.6.5 Causal Modeling

Inspired by the success of causal modeling [166], studying fairness from the causal perspective [135], [136], [137], [138], [139] has attracted increasing attentions. In general, fairness is formulated as the causal effect of the sensitive attribute, which is evaluated by applying counterfactual interventions
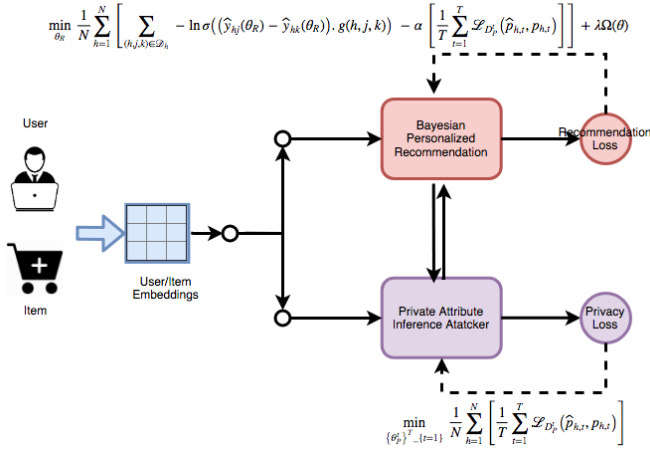
5. https://arxiv.org/pdf/1911.09872.pdf

$$\min_{\theta_R} \frac{1}{N} \sum_{h=1}^{N} \left[\sum_{(h,j,k)\in\mathcal{D}_h} -\ln\sigma\left((\widehat{y}_{hj}(\theta_R) - \widehat{y}_{hk}(\theta_R))\cdot g(h,j,k)\right) - \alpha\left[\frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_{D'_p}(\widehat{p}_{h,t}, p_{h,t})\right]\right] + \lambda\Omega(\theta)$$

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

Fig. 7. Recommendation with attribute protection (RAP) [134]. The figure was reproduced from the Arxiv[5] with authors' permission.

over a causal graph. For example, Wu *et al.* [137] focused on fairness-aware ranking, and argued that the fairness constraints based 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.* [138] introduced the notion of counterfactual fairness, which is derived from Pearl's causal model [166]. 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.

4.7 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 [29], [60]. 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.* [140] 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.* [141] and Bonner *et al.*

[142] 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.* [29] leverage knowledge distillation to extract information from uniform data to learn a unbiased recommendation model. Yu *et al.* [143] leverage influence function to reweight training instances so that it has less loss in an unbiased validation set.

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 8 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 the system [13]. Some recent work [144], [145], [146], [147] 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 [148], [149], [150], [151]. Also, the actor network has been adopted recently to learn the best policy by maximizing the long term reward [152], [153], [154], [155].

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 [156]. 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.* [157] proposes to weight the policy gradient with the inverse of the probability of historical policy. Inspired by [157], some work [156], [158], [159] further explore off-policy evaluation for non-stationary recommendation environments or slate recommendation. However, as claimed by Jeunen *et al.* [160], existing off-policy learning methods usually fail due to stochastic and sparse rewards. Therefore, they [160] further propose to leverage supervised signal with IPS
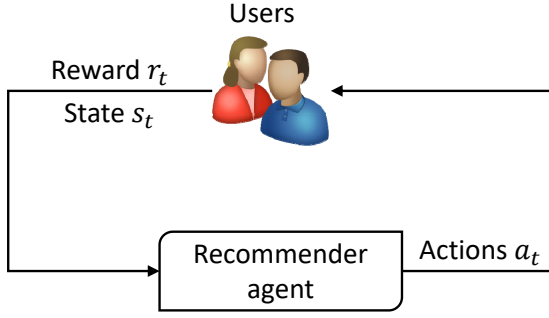
Fig. 8. The system-user interactions with a RL aggent. The figure is plotted referring to [150] with permission.

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.* [161] leverage blind spot term to let items be close to each other in the latent space. Sinha *et al.* [162] 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 debiasing 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 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 140 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. 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] R. Cañamares and P. Castells, "Should i follow the crowd?: A probabilistic analysis of the effectiveness of popularity in recommender systems," in *SIGIR*. ACM, 2018, pp. 415–424.

[2] M. Morik, A. Singh, J. Hong, and T. Joachims, "Controlling fairness and bias in dynamic learning-to-rank," in *SIGIR*. ACM, 2020, pp. 429–438.

[3] Y. Saito, "Asymmetric tri-training for debiasing missing-not-at-random explicit feedback," in *SIGIR*, 2020, pp. 309–318.

[4] J. M. Hernández-Lobato, N. Houlsby, and Z. Ghahramani, "Probabilistic matrix factorization with non-random missing data." in *ICML*, 2014, pp. 1512–1520.

[5] X. Wang, M. Bendersky, D. Metzler, and M. Najork, "Learning to rank with selection bias in personal search," in *SIGIR*, 2016, pp. 115–124.

[6] Z. Ovaisi, R. Ahsan, Y. Zhang, K. Vasilaky, and E. Zheleva, "Correcting for selection bias in learning-to-rank systems," in *Proceedings of The Web Conference 2020*, 2020, pp. 1863–1873.

[7] D. Liang, L. Charlin, J. McInerney, and D. M. Blei, "Modeling user exposure in recommendation," in *WWW*. ACM, 2016, pp. 951–961.

[8] J. Chen, Y. Feng, M. Ester, S. Zhou, C. Chen, and C. Wang, "Modeling users' exposure with social knowledge influence and consumption influence for recommendation," in *CIKM*, 2018, pp. 953–962.

[9] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T. Chua, "Reinforced negative sampling over knowledge graph for recommendation," in *www*. ACM / IW3C2, 2020, pp. 99–109.

[10] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Found. Trends Inf. Retr.*, vol. 14, no. 1, pp. 1–101, 2020.

[11] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artificial intelligence review*, vol. 50, no. 1, pp. 21–48, 2018.

[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[13] X. Zhao, L. Xia, J. Tang, and D. Yin, "Deep reinforcement learning for search, recommendation, and online advertising: a survey," *ACM SIGWEB Newsletter*, no. Spring, pp. 1–15, 2019.

[14] S. L. Blodgett, S. Barocas, H. Daumé III, and H. Wallach, "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*, 2020, pp. 5454–5476.

[15] F. Vella, "Estimating models with sample selection bias: a survey," *Journal of Human Resources*, pp. 127–169, 1998.

[16] X. Zhang and M. Liu, "Fairness in learning-based sequential decision algorithms: A survey," *arXiv preprint arXiv:2001.04861*, 2020.

[17] E. del Barrio, P. Gordaliza, and J.-M. Loubes, "Review of mathematical frameworks for fairness in machine learning," *arXiv preprint arXiv:2005.13755*, 2020.

[18] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[19] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis *et al.*, "Bias in data-driven artificial intelligence systems—an introductory survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1356, 2020.

[20] B. M. Marlin, R. S. Zemel, S. Roweis, and M. Slaney, "Collaborative filtering and the missing at random assumption," in *UAI*, 2007, pp. 267–275.

[21] H. Steck, "Evaluation of recommendations: rating-prediction and ranking," in *RecSys*, 2013, pp. 213–220.

[22] T. Wang and D. Wang, "Why amazon's ratings might mislead you: The story of herding effects," *Big data*, vol. 2, no. 4, pp. 196–204, 2014.

[23] Y. Liu, X. Cao, and Y. Yu, "Are you influenced by others when rating?: Improve rating prediction by conformity modeling," in *RecSys*. ACM, 2016, pp. 269–272.

[24] S. Krishnan, J. Patel, M. J. Franklin, and K. Goldberg, "A methodology for learning, analyzing, and mitigating social influence bias in recommender systems," in *RecSys*, 2014, pp. 137–144.

[25] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *SIGIR*. ACM, 2009, pp. 203–210.

[26] J. Tang, H. Gao, and H. Liu, "mtrust: discerning multi-faceted trust in a connected world," in *WSDM*. ACM, 2012, pp. 93–102.

[27] A. J. Chaney, D. M. Blei, and T. Eliassi-Rad, "A probabilistic model for using social networks in personalized item recommendation," in *RecSys*. ACM, 2015, pp. 43–50.

[28] X. Wang, S. C. Hoi, M. Ester, J. Bu, and C. Chen, "Learning personalized preference of strong and weak ties for social recommendation," in *WWW*. IW3C2, 2017, pp. 1601–1610.

[29] D. Liu, P. Cheng, Z. Dong, X. He, W. Pan, and Z. Ming, "A general knowledge distillation framework for counterfactual recommendation via uniform data," in *SIGIR*, 2020, pp. 831–840.

[30] J. Chen, C. Wang, S. Zhou, Q. Shi, Y. Feng, and C. Chen, "Samwalker: Social recommendation with informative sampling strategy," in *The World Wide Web Conference*. ACM, 2019, pp. 228–239.

[31] J. Chen, C. Wang, S. Zhou, Q. Shi, J. Chen, Y. Feng, and C. Chen, "Fast adaptively weighted matrix factorization for recommendation with implicit feedback." in *AAAI*, 2020, pp. 3470–3477.

[32] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling user interest and popularity bias for recommendation with causal embedding," *CoRR*, vol. abs/2006.11011, 2020.

[33] A. Collins, D. Tkaczyk, A. Aizawa, and J. Beel, "A study of position bias in digital library recommender systems," *arXiv preprint arXiv:1802.06565*, 2018.

[34] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, "Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search," *ACM Transactions on Information Systems (TOIS)*, vol. 25, no. 2, pp. 7–es, 2007.

[35] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *ACM SIGIR Forum*, vol. 51, no. 1. Acm New York, NY, USA, 2017, pp. 4–11.

[36] M. O'Brien and M. T. Keane, "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, 2006, pp. 1881–1886.

[37] K. Klöckner, N. Wirschum, and A. Jameson, "Depth-and breadth-first processing of search result lists," in *CHI'04 extended abstracts on Human factors in computing systems*, 2004, pp. 1539–1539.

[38] C. C. Johnson, "Logistic matrix factorization for implicit feedback data," *NeurIPS*, vol. 27, 2014.

[39] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*. ACM, 2017, pp. 173–182.

[40] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *WSDM*. ACM, 2014, pp. 273–282.

[41] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "Irgan: A minimax game for unifying generative and discriminative information retrieval models," in *SIGIR*. ACM, 2017, pp. 515–524.

[42] D. H. Park and Y. Chang, "Adversarial sampling and training for semi-supervised information retrieval," in *The World Wide Web Conference*. ACM, 2019, pp. 1443–1453.

[43] J. Ding, Y. Quan, X. He, Y. Li, and D. Jin, "Reinforced negative sampling for recommendation with exposure data," in *IJCAI*. AAAI Press, 2019, pp. 2230–2236.

[44] H. Zhang, F. Shen, W. Liu, X. He, H. Luan, and T.-S. Chua, "Discrete collaborative filtering," in *SIGIR*. ACM, 2016, pp. 325–334.

[45] D. Lian, R. Liu, Y. Ge, K. Zheng, X. Xie, and L. Cao, "Discrete content-aware matrix factorization," in *KDD*. ACM, 2017, pp. 325–334.

[46] K. Zhou and H. Zha, "Learning binary codes for collaborative filtering," in *KDD*, 2012, pp. 498–506.

[47] H. Abdollahpouri and M. Mansoury, "Multi-sided exposure bias in recommendation," *arXiv preprint arXiv:2006.15772*, 2020.

[48] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[49] C. C. Aggarwal, "Neighborhood-based collaborative filtering," in *Recommender systems*. Springer, 2016, pp. 29–70.

[50] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *CoRR*, vol. abs/1908.09635, 2019.

[51] B. Friedman and H. Nissenbaum, "Bias in computer systems," *ACM Trans. Inf. Syst.*, vol. 14, no. 3, pp. 330–347, 1996.

[52] A. Lambrecht and C. Tucker, "Algorithmic bias? an empirical study of apparent gender-based discrimination in the display of stem career ads," *Management Science*, vol. 65, no. 7, pp. 2966–2981, 2019.

[53] A. Datta, M. C. Tschantz, and A. Datta, "Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination," *Proceedings on privacy enhancing technologies*, vol. 2015, no. 1, pp. 92–112, 2015.

[54] A. Stoica, C. J. Riederer, and A. Chaintreau, "Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity," in *WWW*, 2018, pp. 923–932.

[55] F. Karimi, M. Génois, C. Wagner, P. Singer, and M. Strohmaier, "Homophily influences ranking of minorities in social networks," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[56] M. D. Ekstrand, M. Tian, M. R. I. Kazi, H. Mehrpouyan, and D. Kluver, "Exploring author gender in book rating and recommendation," in *RecSys*, 2018, pp. 242–250.

[57] M. Mansoury, H. Abdollahpouri, M. Pechenizkiy, B. Mobasher, and R. Burke, "Feedback loop and bias amplification in recommender systems," *arXiv preprint arXiv:2007.13019*, 2020.

[58] A. J. Chaney, B. M. Stewart, and B. E. Engelhardt, "How algorithmic confounding in recommendation systems increases homogeneity and decreases utility," in *RecSys*, 2018, pp. 224–232.

[59] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac, "What recommenders recommend: an analysis of recommendation biases and possible countermeasures," *User Modeling and User-Adapted Interaction*, vol. 25, no. 5, pp. 427–491, 2015.

[60] R. Jiang, S. Chiappa, T. Lattimore, A. György, and P. Kohli, "Degenerate feedback loops in recommender systems," in *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 383–390.

[61] Y. Ge, S. Zhao, H. Zhou, C. Pei, F. Sun, W. Ou, and Y. Zhang, "Understanding echo chambers in e-commerce recommender systems," in *SIGIR*, 2020, pp. 2261–2270.

[62] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, "Recommendations as treatments: Debiasing learning and evaluation," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 48, 2016, pp. 1670–1679.

[63] H. Steck, "Training and testing of recommender systems on data missing not at random," in *KDD*, 2010, pp. 713–722.

[64] B. M. Marlin and R. S. Zemel, "Collaborative prediction and ranking with non-random missing data," in *RecSys*, 2009, pp. 5–12.

[65] J. Chen, C. Wang, M. Ester, Q. Shi, Y. Feng, and C. Chen, "Social recommendation with missing not at random data," in *ICDM*. IEEE, 2018, pp. 29–38.

[66] Y. Wang, D. Liang, L. Charlin, and D. M. Blei, "The deconfounded recommender: A causal inference approach to recommendation," *arXiv preprint arXiv:1808.06581*, 2018.

[67] X. Wang, R. Zhang, Y. Sun, and J. Qi, "Doubly robust joint learning for recommendation on data missing not at random," in *ICML*, 2019, pp. 6638–6647.

[68] L. Yang, Y. Cui, Y. Xuan, C. Wang, S. Belongie, and D. Estrin, "Unbiased offline recommender evaluation for missing-not-at-random implicit feedback," in *RecSys*, 2018, pp. 279–287.

[69] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*. Ieee, 2008, pp. 263–272.

[70] R. Devooght, N. Kourtellis, and A. Mantrach, "Dynamic matrix factorization with priors on unknown values," in *KDD*, 2015, pp. 189–198.

[71] R. Pan and M. Scholz, "Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering," in *KDD*, 2009, pp. 667–676.

[72] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," *ICDM*, pp. 502–511, 2008.

[73] X. He, H. Zhang, M.-Y. Kan, and T.-S. Chua, "Fast matrix factorization for online recommendation with implicit feedback," in *SIGIR*. ACM, 2016, pp. 549–558.

[74] H.-F. Yu, M. Bilenko, and C.-J. Lin, "Selection of negative samples for one-class matrix factorization," in *SDM*. SIAM, 2017, pp. 363–371.

[75] Y. Li, J. Hu, C. Zhai, and Y. Chen, "Improving one-class collaborative filtering by incorporating rich user information," in *CIKM*, 2010, pp. 959–968.

[76] Y. Saito, S. Yaginuma, Y. Nishino, H. Sakata, and K. Nakata, "Unbiased recommender learning from missing-not-at-random implicit feedback," in *WSDM*, 2020, pp. 501–509.

[77] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009, pp. 452–461.

[78] J. Ding, F. Feng, X. He, G. Yu, Y. Li, and D. Jin, "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, 2018, pp. 13–14.

[79] X. Wang, Y. Xu, X. He, Y. Cao, M. Wang, and T.-S. Chua, "Reinforced negative sampling over knowledge graph for recommendation," in *Proceedings of The Web Conference 2020*, 2020, pp. 99–109.

[80] M. Wang, X. Zheng, Y. Yang, and K. Zhang, "Collaborative filtering with social exposure: A modular approach to social recommendation," in *AAAI*, 2018.

[81] W. Zhang, W. Bao, X.-Y. Liu, K. Yang, Q. Lin, H. Wen, and R. Ramezani, "Large-scale causal approaches to debiasing post-click conversion rate estimation with multi-task learning," in *Proceedings of The Web Conference 2020*, 2020, pp. 2775–2781.

[82] X. Ma, L. Zhao, G. Huang, Z. Wang, Z. Hu, X. Zhu, and K. Gai, "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *SIGIR*, 2018, pp. 1137–1140.

[83] H. Wen, J. Zhang, Y. Wang, F. Lv, W. Bao, Q. Lin, and K. Yang, "Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction," in *SIGIR*, 2020, pp. 2377–2386.

[84] W. Bao, H. Wen, S. Li, X.-Y. Liu, Q. Lin, and K. Yang, "Gmcm: Graph-based micro-behavior conversion model for post-click conversion rate estimation," in *SIGIR*, 2020, pp. 2201–2210.

[85] W. Wang, F. Feng, X. He, H. Zhang, and T.-S. Chua, """ click" is not equal to" like": Counterfactual recommendation for mitigating clickbait issue," *arXiv preprint arXiv:2009.09945*, 2020.

[86] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey, "An experimental comparison of click position-bias models," in *WSDM*, 2008, pp. 87–94.

[87] G. E. Dupret and B. Piwowarski, "A user browsing model to predict search engine click data from past observations." in *SIGIR*, 2008, pp. 331–338.

[88] O. Chapelle and Y. Zhang, "A dynamic bayesian network click model for web search ranking," in *WWW*, 2009, pp. 1–10.

[89] W. V. Zhang and R. Jones, "Comparing click logs and editorial labels for training query rewriting," in *WWW 2007 Workshop on Query Log Analysis: Social And Technological Challenges*, 2007.

[90] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos, "Click chain model in web search," in *WWW*, 2009, pp. 11–20.

[91] Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen, "A novel click model and its applications to online advertising," in *WSDM*, 2010, pp. 321–330.

[92] J. Jin, Y. Fang, W. Zhang, K. Ren, G. Zhou, J. Xu, Y. Yu, J. Wang, X. Zhu, and K. Gai, "A deep recurrent survival model for unbiased ranking," *arXiv preprint arXiv:2004.14714*, 2020.

[93] A. Agarwal, K. Takatsu, I. Zaitsev, and T. Joachims, "A general framework for counterfactual learning-to-rank," in *SIGIR*, 2019, pp. 5–14.

[94] T. Joachims, A. Swaminathan, and T. Schnabel, "Unbiased learning-to-rank with biased feedback," in *WSDM*, 2017, pp. 781–789.

[95] A. Schuth, H. Oosterhuis, S. Whiteson, and M. de Rijke, "Multileave gradient descent for fast online learning to rank," in *WSDM*, 2016, pp. 457–466.

[96] K. Raman and T. Joachims, "Learning socially optimal information systems from egoistic users," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 128–144.

[97] A. Swaminathan and T. Joachims, "Batch learning from logged bandit feedback through counterfactual risk minimization," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1731–1755, 2015.

[98] K. Hofmann, A. Schuth, S. Whiteson, and M. De Rijke, "Reusing historical interaction data for faster online learning to rank for ir," in *WSDM*, 2013, pp. 183–192.

[99] Z. Fang, A. Agarwal, and T. Joachims, "Intervention harvesting for context-dependent examination-bias estimation," in *SIGIR*, 2019, pp. 825–834.

[100] A. Agarwal, I. Zaitsev, X. Wang, C. Li, M. Najork, and T. Joachims, "Estimating position bias without intrusive interventions," in *WSDM*, 2019, pp. 474–482.

[101] Q. Ai, K. Bi, C. Luo, J. Guo, and W. B. Croft, "Unbiased learning to rank with unbiased propensity estimation," in *SIGIR*, 2018, pp. 385–394.

[102] X. Wang, N. Golbandi, M. Bendersky, D. Metzler, and M. Najork, "Position bias estimation for unbiased learning to rank in personal search," in *WSDM*, 2018, pp. 610–618.

[103] A. Vardasbi, M. de Rijke, and I. Markov, "Cascade model-based propensity estimation for counterfactual learning to rank," in *SIGIR*, 2020, pp. 2089–2092.

[104] Z. Qin, S. J. Chen, D. Metzler, Y. Noh, J. Qin, and X. Wang, "Attribute-based propensity for unbiased learning in recommender systems: Algorithm and case studies," in *KDD*, 2020, pp. 2359–2367.

[105] R. Guo, X. Zhao, A. Henderson, L. Hong, and H. Liu, "Debiasing grid-based product search in e-commerce," in *KDD*, 2020, pp. 2852–2860.

[106] H. Abdollahpouri, R. Burke, and B. Mobasher, "Controlling popularity bias in learning-to-rank recommendation," in *RecSys*, 2017, pp. 42–46.

[107] J. Wasilewski and N. Hurley, "Incorporating diversity in a learning to rank recommender system," in *FLAIRS*, 2016, pp. 572–578.

[108] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Correcting popularity bias by enhancing recommendation neutrality," in *RecSys*, 2014.

[109] Z. Chen, R. Xiao, C. Li, G. Ye, H. Sun, and H. Deng, "ESAM: discriminative domain adaptation with non-displayed items to improve long-tail performance," in *SIGIR*, 2020, pp. 579–588.

[110] A. Krishnan, A. Sharma, A. Sankar, and H. Sundaram, "An adversarial approach to improve long-tail performance in neural collaborative filtering," in *CIKM*, 2018, pp. 1491–1494.

[111] M. Bressan, S. Leucci, A. Panconesi, P. Raghavan, and E. Terolli, "The limits of popularity-based recommendations, and the role of social ties," in *KDD*, 2016, pp. 745–754.

[112] H. Abdollahpouri, "Popularity bias in ranking and recommendation," in *AIES*, 2019, pp. 529–530.

[113] D. Pedreschi, S. Ruggieri, and F. Turini, "Discrimination-aware data mining," in *KDD*, 2008, pp. 560–568.

[114] S. C. Geyik, S. Ambler, and K. Kenthapadi, "Fairness-aware ranking in search & recommendation systems with application to linkedin talent search," in *KDD*, 2019, pp. 2221–2231.

[115] A. Asudeh, H. V. Jagadish, J. Stoyanovich, and G. Das, "Designing fair ranking schemes," in *SIGMOD*, 2019, pp. 1259–1276.

[116] A. J. Biega, K. P. Gummadi, and G. Weikum, "Equity of attention: Amortizing individual fairness in rankings," in *SIGIR*, 2018, pp. 405–414.

[117] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "Fa*ir: A fair top-k ranking algorithm," in *CIKM*, 2017, pp. 1569–1578.

[118] A. Singh and T. Joachims, "Fairness of exposure in rankings," in *KDD*, 2018, pp. 2219–2228.

[119] G. Farnadi, P. Kouki, S. K. Thompson, S. Srinivasan, and L. Getoor, "A fairness-aware hybrid recommender system," *CoRR*, vol. abs/1809.09030, 2018.

[120] T. A. Rahman, B. Surma, M. Backes, and Y. Zhang, "Fairwalk: Towards fair graph embedding," in *IJCAI*, 2019, pp. 3289–3295.

[121] M. Buyl and T. D. Bie, "Debayes: a bayesian method for debiasing network embeddings," *CoRR*, vol. abs/2002.11442, 2020.

[122] R. S. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *ICML*, vol. 28, 2013, pp. 325–333.

[123] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Enhancement of the neutrality in recommendation," in *RecSys*, vol. 893, 2012, pp. 8–14.

[124] ——, "Efficiency improvement of neutrality-enhanced recommendation." in *Decisions@ RecSys*, 2013, pp. 1–8.

[125] T. Kamishima, S. Akaho, H. Asoh, and I. Sato, "Model-based approaches for independence-enhanced recommendation," in *ICDM*, 2016, pp. 860–867.

[126] T. Kamishima and S. Akaho, "Considerations on recommendation independence for a find-good-items task," in *FATREC Workshop on Responsible Recommendation Proceedings*, 2017.

[127] S. Yao and B. Huang, "Beyond parity: Fairness objectives for collaborative filtering," in *NeurIPS*, 2017, pp. 2921–2930.

[128] ——, "New fairness metrics for recommendation that embrace differences," *CoRR*, vol. abs/1706.09838, 2017.

[129] X. Lin, M. Zhang, Y. Zhang, Z. Gu, Y. Liu, and S. Ma, "Fairness-aware group recommendation with pareto-efficiency," in *RecSys*, 2017, pp. 107–115.

[130] R. Burke, N. Sonboli, M. Mansoury, and A. Ordoñez-Gauger, "Balanced neighborhoods for fairness-aware collaborative recommendation," in *FATREC Workshop on Responsible Recommendation Proceedings*, 2017, p. 5.

[131] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi, and C. Goodrow, "Fairness in recommendation ranking through pairwise comparisons," in *KDD*, 2019, pp. 2212–2220.

[132] H. Edwards and A. J. Storkey, "Censoring representations with an adversary," in *ICLR*, 2016.

[133] A. J. Bose and W. L. Hamilton, "Compositional fairness constraints for graph embeddings," in *ICML*, vol. 97, 2019, pp. 715–724.

[134] G. Beigi, A. Mosallanezhad, R. Guo, H. Alvari, A. Nou, and H. Liu, "Privacy-aware recommendation with private-attribute protection using adversarial learning," in *WSDM*, 2020, pp. 34–42.

[135] J. Zhang and E. Bareinboim, "Fairness in decision-making - the causal explanation formula," in *AAAI*, 2018, pp. 2037–2045.

[136] R. Nabi and I. Shpitser, "Fair inference on outcomes," in *AAAI*, 2018, pp. 1931–1940.

[137] Y. Wu, L. Zhang, and X. Wu, "On discrimination discovery and removal in ranked data using causal graph," in *KDD*. ACM, 2018, pp. 2536–2544.

[138] M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva, "Counterfactual fairness," in *NeurIPS*, 2017, pp. 4066–4076.

[139] Y. Wu, L. Zhang, and X. Wu, "Counterfactual fairness: Unidentification, bound and algorithm." in *IJCAI*, 2019, pp. 1438–1444.

[140] B. Yuan, J.-Y. Hsia, M.-Y. Yang, H. Zhu, C.-Y. Chang, Z. Dong, and C.-J. Lin, "Improving ad click prediction by considering non-displayed events," in *CIKM*, 2019, pp. 329–338.

[141] N. Rosenfeld, Y. Mansour, and E. Yom-Tov, "Predicting counterfactuals from large historical data and small randomized trials," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 602–609.

[142] S. Bonner and F. Vasile, "Causal embeddings for recommendation," in *RecSys*, 2018, pp. 104–112.

[143] J. Yu, H. Zhu, C.-Y. Chang, X. Feng, B. Yuan, X. He, and Z. Dong, "Influence function for unbiased recommendation," in *SIGIR*, 2020, pp. 1929–1932.

[144] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *WWW*, 2010, pp. 661–670.

[145] X. Wang, S. C. Hoi, C. Liu, and M. Ester, "Interactive social recommendation," in *CIKM*, 2017, pp. 357–366.

[146] H. Wang, Q. Wu, and H. Wang, "Factorization bandits for interactive recommendation." in *AAAI*, vol. 17, 2017, pp. 2695–2702.

[147] X. Zhao, W. Zhang, and J. Wang, "Interactive collaborative filtering," in *CIKM*, 2013, pp. 1411–1420.

[148] X. Zhao, X. Zheng, X. Yang, X. Liu, and J. Tang, "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*, 2020, pp. 3319–3327.

[149] S.-Y. Chen, Y. Yu, Q. Da, J. Tan, H.-K. Huang, and H.-H. Tang, "Stabilizing reinforcement learning in dynamic environment with application to online recommendation," in *KDD*, 2018, pp. 1187–1196.

[150] X. Zhao, L. Zhang, Z. Ding, L. Xia, J. Tang, and D. Yin, "Recommendations with negative feedback via pairwise deep reinforcement learning," in *KDD*, 2018, pp. 1040–1048.

[151] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 167–176.

[152] H. Chen, X. Dai, H. Cai, W. Zhang, X. Wang, R. Tang, Y. Zhang, and Y. Yu, "Large-scale interactive recommendation with tree-structured policy gradient," in *AAAI*, vol. 33, 2019, pp. 3312–3320.

[153] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for page-wise recommendations," in *RecSys*, 2018, pp. 95–103.

[154] X. Zhao, L. Zhang, L. Xia, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for list-wise recommendations," *arXiv preprint arXiv:1801.00209*, 2017.

[155] X. Wang, Y. Chen, J. Yang, L. Wu, Z. Wu, and X. Xie, "A reinforcement learning framework for explainable recommendation," in *ICDM*. IEEE, 2018, pp. 587–596.

[156] R. Jagerman, I. Markov, and M. de Rijke, "When people change their mind: Off-policy evaluation in non-stationary recommendation environments," in *WSDM*, 2019, pp. 447–455.

[157] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi, "Top-k off-policy correction for a reinforce recommender system," in *WSDM*, 2019, pp. 456–464.

[158] J. McInerney, B. Brost, P. Chandar, R. Mehrotra, and B. Carterette, "Counterfactual evaluation of slate recommendations with sequential reward interactions," in *KDD*, 2020, pp. 1779–1788.

[159] A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni, "Off-policy evaluation for slate recommendation," in *Advances in Neural Information Processing Systems*, 2017, pp. 3632–3642.

[160] O. Jeunen, D. Rohde, F. Vasile, and M. Bompaire, "Joint policy-value learning for recommendation," in *KDD*, 2020, pp. 1223–1233.

[161] W. Sun, S. Khenissi, O. Nasraoui, and P. Shafto, "Debiasing the human-recommender system feedback loop in collaborative filtering," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 645–651.

[162] A. Sinha, D. F. Gleich, and K. Ramani, "Deconvolving feedback loops in recommender systems," in *Advances in neural information processing systems*, 2016, pp. 3243–3251.

[163] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*, 2016, pp. 785–794.

[164] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *WSDM*. ACM, 2016, pp. 153–162.

[165] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*, 2020, pp. 639–648.

[166] J. Pearl, *Causality*. Cambridge university press, 2009.

[167] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, "Efficiency improvement of neutrality-enhanced recommendation," in *RecSys*, 2013, pp. 1–8.

[168] N. Grgic-Hlaca, M. B. Zafar, K. P. Gummadi, and A. Weller, "The case for process fairness in learning: Feature selection for fair

decision making," in *NeurIPS Symposium on Machine Learning and the Law*, vol. 1, 2016, p. 2.

[169] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, "Fairness through awareness," in *Innovations in Theoretical Computer Science*, 2012, pp. 214–226.

[170] M. Joseph, M. J. Kearns, J. Morgenstern, S. Neel, and A. Roth, "Rawlsian fairness for machine learning," *CoRR*, vol. abs/1610.09559, 2016.

[171] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. S. Zemel, "The variational fair autoencoder," in *ICLR*, 2016.

[172] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *AISTATS*, vol. 54, 2017, pp. 962–970.

[173] J. S. Kim, J. Chen, and A. Talwalkar, "Fact: A diagnostic for group fairness trade-offs," *ICML*, 2020.

[174] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *NeurIPS*, 2016, pp. 3315–3323.

[175] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment," in *WWW*, 2017, pp. 1171–1180.

[176] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big Data*, vol. 5, no. 2, pp. 153–163, 2017.

[177] J. M. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," in *ITCS*, ser. LIPIcs, vol. 67, 2017, pp. 43:1–43:23.

[178] R. Burke, "Multisided fairness for recommendation," *CoRR*, vol. abs/1707.00093, 2017.

[179] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz, "Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems," in *CIKM*, 2018, pp. 2243–2251.

[180] W. Liu, J. Guo, N. Sonboli, R. Burke, and S. Zhang, "Personalized fairness-aware re-ranking for microlending," in *RecSys*, 2019, pp. 467–471.

[181] Z. Zhu, X. Hu, and J. Caverlee, "Fairness-aware tensor-based recommendation," in *CIKM*, 2018, pp. 1153–1162.

Xiang Wang is now a research fellow at National University of Singapore. He received his Ph.D. degree from National University of Singapore in 2019. His research interests include recommender systems, graph learning, and explainable deep learning techniques. He has published some academic papers on international conferences such as KDD, WWW, SIGIR, and AAAI. He serves as a program committee member for several top conferences such as SIGIR and WWW.
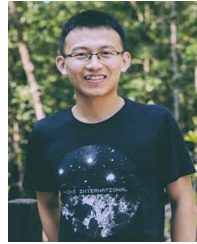
Fuli Feng is a Research Fellow in the School of Computing, National University of Singapore (NUS). He received Ph.D. in Computer Science from NUS in 2019. His research interests include information retrieval, data mining, and multi-media processing. He has over 30 publications appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE and TOIS. His work on Bayesian Personalized Ranking has received the Best Poster Award of WWW 2018. Moreover, he has been served as the PC member for several top conferences including SIGIR, WWW, WSDM, NeurIPS, AAAI, ACL, MM, and invited reviewer for prestigious journals such as TOIS, TKDE, TNNLS, TPAMI, and TMM.

Jiawei Chen is a Postdoc Research Fellow in School of Information Science and Technology, University of Science and Technology of China. He received Ph.D. in Computer Science from Zhejiang University in 2020. His research interests include information retrieval, data mining, and causal inference. He has published some academic papers on international conferences such as WWW, AAAI, CIKM, ICDE and ICDM. Moreover, he has served as the PC member for top-tier conferences including SIGIR, ACMMM, AAAI, IJCAI and the invited reviewer for prestigious journals such as TNNLS, TKDE, TOIS.

Meng Wang is a professor at the Hefei University of Technology, China. He received his B.E. degree and Ph.D. degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2008, respectively. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored more than 200 book chapters, journal and conference papers in these areas. He is the recipient of the ACM SIGMM Rising Star Award 2014. He is an associate editor of IEEE TKDE, TCSVT, and TNNLS.

Hande Dong is pursuing his master degree in the School of Information Science and Technology, University of Science and Technology of China (USTC). His current research interest focuses on machine learning and data mining.

Xiangnan He is a professor at the University of Science and Technology of China (USTC). His research interests span information retrieval, data mining, and multi-media analytics. He has over 90 publications in top conferences such as SIGIR, WWW, and MM, KDD, and journals including TKDE, TOIS, and TMM. His work has received the Best Paper Award Honorable Mention in WWW 2018 and SIGIR 2016. He is in the Editorial Board of the AI Open journal, served as the PC chair of CCIS 2019, the area chair of MM 2019, ECML-PKDD 2020, and the (senior) PC member for top conferences including SIGIR, WWW, KDD, WSDM etc.