Uncertainty Quantification for Fairness in Two-Stage Recommender Systems

Lequn Wang
Cornell University
Ithaca, NY, USA
lw633@cornell.edu

Thorsten Joachims
Cornell University
Ithaca, NY, USA
tj@cs.cornell.edu

ABSTRACT

Many large-scale recommender systems consist of two stages. The first stage efficiently screens the complete pool of items for a small subset of promising candidates, from which the second-stage model curates the final recommendations. In this paper, we investigate how to ensure group fairness to the items in this two-stage architecture. In particular, we find that existing first-stage recommenders might select an irrecoverably unfair set of candidates such that there is no hope for the second-stage recommender to deliver fair recommendations. To this end, motivated by recent advances in uncertainty quantification, we propose two threshold-policy selection rules that can provide distribution-free and finite-sample guarantees on fairness in first-stage recommenders. More concretely, given any relevance model of queries and items and a point-wise lower confidence bound on the expected number of relevant items for each threshold-policy, the two rules find near-optimal sets of candidates that contain enough relevant items in expectation from each group of items. To instantiate the rules, we demonstrate how to derive such confidence bounds from potentially partial and biased user feedback data, which are abundant in many large-scale recommender systems. In addition, we provide both finite-sample and asymptotic analyses of how close the two threshold selection rules are to the optimal thresholds. Beyond this theoretical analysis, we show empirically that these two rules can consistently select enough relevant items from each group while minimizing the size of the candidate sets for a wide range of settings.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Two-Stage Recommender Systems, Distribution-Free Uncertainty Quantification, Algorithmic Fairness

ACM Reference Format:
Lequn Wang and Thorsten Joachims. 2023. Uncertainty Quantification for Fairness in Two-Stage Recommender Systems. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570469

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570469

1 INTRODUCTION

Two-stage pipelines [6, 11, 15, 21, 27, 72, 75, 76] are ubiquitous in large-scale recommender systems. Their key advantage lies in their efficiency and scalability, making it possible to curate personalized recommendations from billions of items within milliseconds [44]. The first stage focuses on efficiently generating a small set of candidates that contains enough relevant items. To achieve the necessary efficiency, models used in the first stage may be less accurate and biased. The second stage only considers the candidates selected in the first stage for generating the final recommendations. It can thus be more resource intensive, which allows second-stage models to be more accurate and less biased.

While much prior work has focused on improving the efficiency and overall effectiveness of two-stage pipelines [15, 30, 38, 44], less attention has been given to the fairness aspects of this two-stage architecture. We therefore investigate methods for ensuring fair allocation of exposure to the items and their providers in two-stage pipelines, for which there are ample ethical [39], economical (*e.g.*, provider retention, super-star economics [45]), and legal (*e.g.*, anti-trust law [56]) reasons. We specifically investigate how the first stage impacts the fairness of the recommendations, making our work complementary to the existing body of work on fairness and diversity of the second stage in bandits [12, 24, 36, 47, 55, 63] and rankings [4, 5, 9, 18, 19, 23, 28, 33, 40, 41, 45, 48, 51, 57, 64, 69, 74]. These second-stage methods do not apply to the first stage, since their computation overhead is at least linear in the number of items, which would lead to unacceptable latency in the first stage[1].

We consider a group-based notion of fairness to the items. Since the second-stage recommender makes the final recommendations from the candidate set produced by the first stage, a key requirement for the first stage is to select enough relevant items from each group of items to avoid generating an irrecoverably unfair set of candidates. For example, consider an e-commerce recommender system, where we aim to ensure both small businesses and large businesses receive an equitable amount of exposure to the users. Without a fairness-aware candidate generation policy in the first stage, it might happen that the group of items belonging to small businesses are disproportionately selected less in the first stage, which might be due to biased relevance estimation towards the items from small businesses. In this case, there is little hope for the second-stage recommendation policy to ensure fairness, since (1)

[1]The first-stage recommenders typically employ approximate algorithms to retrieve approximately top-scored items with sub-linear (in the number of items) time complexity, *e.g.*, locality-sensitive hashing [15, 31, 73].

there might not be enough relevant items from small businesses for the second-stage recommendations to be fair; (2) second-stage recommendation policies typically only ensure fairness proportional to the items selected in the first stage, where small businesses are already unfairly represented. These fairness issues can appear in almost any two-stage recommender system where we need to consider fair allocation of exposure to the items, including those in hiring, online streaming, and social media.

In this paper, we study how to ensure fairness with distribution-free and finite-sample guarantees in the first stage of two-stage recommender systems, while retaining the efficiency of existing first-stage recommender systems. In particular, limited by the latency requirements and motivated by the Rooney rule [13], we focus on constructing threshold-based first-stage candidate generation policies that can provably select the smallest sets of candidates that contain a desired expected number of relevant items from each group, given any—possibly biased—relevance model. These guarantees make our approach different from existing works that rely on reducing the bias of relevance estimation in recommendation policies to improve fairness [8, 52, 71]. While these are certainly useful for the first stage, they do not provide finite-sample and distribution-free guarantees on the fairness and quality of the items selected in the first stage.

Contributions. We formalize fairness objectives for the first stage of recommender pipelines and propose two threshold selection rules—which we call the union rule and the monotone rule—motivated by distribution-free uncertainty quantification methods. We show that, given any relevance model of queries and items, they can provably select a desired number of relevant items in expectation from each group with high probability, while minimizing the candidate-set size. This result holds even if the relevance model is biased against some groups. We also provide both finite-sample and asymptotic analysis on how close the thresholds selected by the two rules are to the optimal ones. The threshold selection rules and the near-optimality analysis rely on lower and upper confidence bounds on the expected number of relevant items from each group for candidate generation policies. Thus, we derive such confidence bounds from potentially biased and partial user feedback data (*e.g.*, user clicks), which are abundant in many recommender systems. From these bounds, we show that the two threshold selection rules approach the optimal thresholds asymptotically. In addition, we also discuss how the proposed first-stage recommendation policy design can shift the cost of inaccurate relevance estimation and lack of data from the disadvantaged groups[2] to the latency of the second-stage recommender, which provides economic incentives for the decision makers to construct more accurate relevance models and to collect more data for every group of items.

Finally, we corroborate the theoretical analysis of the two proposed selection rules with an empirical evaluation on the Microsoft Learning to Rank dataset [50] against several baselines. The results show that only the two proposed selection rules can consistently select enough relevant items from each group across different amounts of user feedback data and accuracies of the relevance model. With a decent amount of data, the two proposed selection rules achieve the smallest candidate-set size among the methods that can select enough relevant items. We also conduct ablation studies to test their robustness to the parameters in the selection rules. The code for the empirical evaluation is accessible at https://github.com/LequnWang/Fair-Two-Stage-Recommender.

[2]Disadvantaged groups in this paper refer to groups of items for which the relevance estimation is inaccurate/biased, and/or we lack user feedback data.

2 FURTHER RELATED WORK

Our proposed threshold selection rules are inspired by distribution-free uncertainty quantification methods [25], including calibration [17, 49] and conformal prediction [62]. The goal of distribution-free uncertainty quantification is to provide point estimates with finite-sample distribution-free error guarantees (calibration) or confidence intervals (conformal prediction) of some target parameters of interest. In this context, the most relevant work is arguably by Bates *et al.* [3], where they provide a strategy to control the risk of prediction sets from a pool of candidates. Our proposed monotone threshold selection rule uses ideas similar to their strategy. The strategy relies on a point-wise lower confidence bound on the risk, which they derive from full-information data. In contrast, we derive the confidence bounds using partial and biased user feedback, which we can typically have easy access to in recommender systems. In addition, we also provide both finite-sample and asymptotic near-optimality analyses of the two proposed threshold selection rules.

The confidence bounds we derive are built upon literature on off-policy evaluation in recommender systems [7, 34, 54, 59]. Many works have proposed estimators to estimate the expected utility of a contextual bandit [20, 37, 58, 60, 68] or a ranking policy [35, 46, 58, 66, 70] from biased and partial user feedback. Some works have derived finite-sample confidence bounds on the estimation error in contextual bandits [7, 42]. We use similar techniques to derive confidence bounds around the clipped inverse propensity weighted estimator [7, 32] for the ranking setting.

Threshold selection rules have also been applied to screening processes [14, 53, 65]. However, these works assume that the candidates are independent and identically distributed (*i.i.d.*). In contrast, we consider recommendation scenarios where the relevances of the items are dependent given a recommendation request. Thus these approaches do not apply here.

3 FAIRNESS IN THE FIRST STAGE

We consider recommendation problems with n items[3] $\boldsymbol{d} = \left(d^j\right)_{j\in[n]}$, where each item d^j belongs to the item space $\mathcal{D}$, *i.e.*, $d^j \in \mathcal{D}$ for all $j \in [n]$. We want to select enough relevant items from several (possibly overlapping) groups $\mathcal{G}$ of items. We assume the group information is known for every item d^j and can be included in the feature representation of the item. We model the distribution of requests coming into the recommender system using a distribution $P_{Q,\boldsymbol{R}}$ over queries and relevance vectors. For each recommendation request, the query $q \in \mathcal{Q}$ and the relevance vector $\boldsymbol{r} = \left(r^j\right)_{j\in[n]}$ of the items are independently drawn from $q, \boldsymbol{r} \sim P_{Q,\boldsymbol{R}}$, where $r^j \in \{0, 1\}$ is the relevance of item d^j to the recommendation request. The first-stage recommender relies on an expected-relevance estimation model[4] $f : \mathcal{Q} \times \mathcal{D} \to [0, 1]$, which maps a query q and an item d^j

[3]We use $[\cdot]$ to denote the set $\{1, 2, \ldots, \cdot\}$.

[4]We call it the relevance model interchangeably.

to an estimate $f\left(q, d^j\right)$ of the expected relevance[5] $\mathbb{E}\left[R^j \mid Q = q\right]$. Given a fixed first-stage relevance model f, a first-stage candidate generation policy $\pi^f : \mathcal{Q} \times \mathcal{D}^n \to \{0,1\}^n$ maps a query q and the items $\left(d^j\right)_{j\in[n]}$ to the selection decisions $\boldsymbol{s} = \pi\left(q, \boldsymbol{d}\right)$, where $\boldsymbol{s} = \left(s^j\right)_{j\in[n]}$ and $s^j \in \{0,1\}$ represents whether the item d^j is selected ($s^j = 1$) or not selected ($s^j = 0$).

Ideally, we would like a policy π^f that selects only items that are relevant to the recommendation request from each group. Unfortunately, as long as there is no deterministic mapping between the query q and the relevance vector $\boldsymbol{r}$, such a perfect candidate generation policy does not exist in general. What is worse, to satisfy the latency requirements, recommender systems require that f be computed efficiently, often at the expense of accuracy and unbiasedness. So instead, we focus on constructing a policy π^f that creates sets of items that are *near-optimal* in terms of the candidate-set size, while provably containing *enough* relevant items in expectation for each group of items, without making assumptions on the distribution $P_{Q,\boldsymbol{R}}$ nor the accuracy/bias of the relevance model f.

In particular, given any relevance model f, we consider group-aware threshold policies that select $t_g \in [t_g^{\max}]$ top-scored (predicted by the relevance model f) items from each group g. $t_g^{\max}$ is the largest candidate-set size the decision makers consider for a group g, which conveys the decision makers' belief that there must be enough relevant items from group g, had they selected $t_g^{\max}$ top-scored items from g. Note that considering only thresholds up to $t_g^{\max}$ provides both computational and statistical efficiency as discussed later. Formally, let $\boldsymbol{t} = \left(t_g\right)_{g\in\mathcal{G}}$, a threshold policy $\pi_{\boldsymbol{t}}^f$ selects an item if it is among the t_g top-scored items from a group g, *i.e.*,

$$s^j = \begin{cases} 1 & \text{if } \exists g \in \mathcal{G} \text{ s.t. } j \in \left\{\sigma_{q,g}^f(j') : j' \in \left[t_g\right]\right\} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\sigma_{q,g}^f$ is a ranking of the items in the group g by their estimated relevance to the query q predicted by f, which returns the index $\sigma_{q,g}^f(j)$ of the item that is ranked at position j from group g . We assume that $\sigma_{q,g}^f$ is deterministic without loss of generality.

We aim to select a threshold vector $\boldsymbol{t} \in \prod_{g\in\mathcal{G}} \left[t_g^{\max}\right]$ such that the expected number of relevant items from each group $g \in \mathcal{G}$ is greater than a target $U_g^\star \in \mathbb{R}$ specified by the decision makers, *i.e.*,

$$U_g\left(t_g\right) \coloneqq U_g\left(\pi_{\boldsymbol{t}}^f\right) = \mathbb{E}_{q,\boldsymbol{r}\sim P_{Q,\boldsymbol{R}}}\left[\sum_{j\in[t_g]} r^{\sigma_{q,g}^f(j)}\right] \geq U_g^\star, \tag{2}$$

while minimizing the candidate-set size t_g of each group g. Note that when the groups are disjoint, minimizing the candidate-set size of each group is equivalent to minimizing the candidate-set size of the whole first-stage recommender. Throughout the paper, we make the following mild assumption on $t_g^{\max}$.

Assumption 3.1. *For any group $g \in \mathcal{G}$, $U_g\left(t_g^{\max}\right) \geq U_g^\star > 0$.*

The target expected numbers of relevant items $\left(U_g^\star\right)_{g\in\mathcal{G}}$ reflect the decision makers' belief that they can build a fair second-stage recommender system given $U_g^\star$ relevant items from each group g.

4 FAIR AND NEAR-OPTIMAL THRESHOLD SELECTION RULES

In this section, we first introduce two threshold selection rules that can provably select enough relevant items in expectation from each group with high probability, while achieving near-optimal candidate-set sizes. Both rules rely on a point-wise lower confidence bound on the expected number of relevant items for each threshold policy and each group, and their near-optimality gaps further depend on a point-wise upper confidence bound on that quantity. Thus, we derive such lower and upper confidence bounds from some user feedback data (*e.g.*, user clicks), which might be partial and biased. From these bounds, we instantiate concrete threshold selection algorithms, provide asymptotic analysis on how close the proposed threshold selection rules are to the optimal, and discuss how these two rules can incentivize the decision makers to improve the accuracy of the relevance model and collect more data for the disadvantaged groups.

4.1 Threshold Selection Rules

For now, lets assume that we have access to a point-wise lower confidence bound $\hat{U}_g^-(t,\alpha)$ on the expected number of relevant items $U_g(t)$ such that for any group $g \in \mathcal{G}$, threshold[6] $t \in \left[0 : t_g^{\max} - 1\right]$, and $\alpha \in (0,1)$,

$$\Pr\left(U_g(t) \geq \hat{U}_g^-(t,\alpha)\right) \geq 1 - \alpha.$$

We will derive one such bound using user feedback data in Section 4.3. Given this bound, we propose two threshold selection rules which can ensure that the expected number of relevant items is above the target level $U_g^\star$ with high probability, while achieving near-optimal expected candidate-set size for a group g.

The first rule we propose is called the *union threshold selection rule*. Given any success probability $1 - \alpha \in (0,1)$, it selects the smallest threshold $\hat{t}_g^{\text{union}}$ for a group g such that the lower confidence bound on the expected number of relevant items with failure probability $\frac{\alpha}{t_g^{\max}-1}$ is greater than the target, *i.e.*,

$$\hat{t}_g^{\text{union}} \coloneqq \min\left\{t \in \left[t_g^{\max} - 1\right] : \hat{U}_g^-\left(t, \frac{\alpha}{\left(t_g^{\max} - 1\right)}\right) \geq U_g^\star\right\}, \tag{3}$$

where we define the minimum over the empty set for a group g to be $t_g^{\max}$, *i.e.*, we set the threshold to be $t_g^{\max}$ if none of the lower bounds exceeds the target. We show that the union threshold selection rule can select enough relevant items for a group g with high probability in the following theorem, with a proof that applies the union bound over the lower confidence bounds for each threshold, as reflected in our naming of the rule.

[5] We use capital letters to denote random variables and lower case letters to denote realizations of random variables.

[6] We use $[0:\cdot]$ to denote the set $\{0, 1, \ldots, \cdot\}$.

Theorem 4.1. *Under Assumption 3.1, for any group $g \in \mathcal{G}$ and $\alpha \in (0,1)$, with probability at least $1-\alpha$,*

$$U_g\left(\hat{t}_g^{\text{union}}\right) \geq U_g^\star.$$

Proof. For a group g, applying the union bound to the lower confidence bounds for each threshold, and by Assumption 3.1, we have that with probability at least $1-\alpha$,

$$U_g(t) \geq \hat{U}_g^-\left(t, \frac{\alpha}{\left(t_g^{\max}-1\right)}\right) \quad \forall t \in \left[t_g^{\max}\right].$$

When the above event holds,

$$U_g\left(\hat{t}_g^{\text{union}}\right) \geq \hat{U}_g^-\left(\hat{t}_g^{\text{union}}, \frac{\alpha}{\left(t_g^{\max}-1\right)}\right) \geq U_g^\star,$$

where the second inequality is by the definition of $\hat{t}_g^{\text{union}}$. ∎

The second selection rule is called the *monotone threshold selection rule*. Given any success probability $1-\alpha \in (0,1)$, it selects the smallest threshold $\hat{t}_g^{\text{mono}}$ for a group g such that the lower confidence bound with failure probability α for every threshold larger than or equal to $\hat{t}_g^{\text{mono}}$ is greater than the target $U_g^\star$, *i.e.*,

$$\hat{t}_g^{\text{mono}} := \min\left\{t \in \left[t_g^{\max}-1\right] : \hat{U}_g^-\left(t', \alpha\right) \geq U_g^\star, \forall t \leq t' < t_g^{\max}\right\}. \quad (4)$$

Compared to $\hat{t}_g^{\text{union}}$, $\hat{t}_g^{\text{mono}}$ allows for a larger failure probability in the confidence lower bounds for each threshold, but requires that the lower bounds be greater than the target for all the thresholds larger than the selected one, in addition to the selected one. It can also ensure selecting enough relevant items with high probability for any group as shown in the following theorem, where the proof leverages the fact that U_g is monotonically increasing, as reflected in our naming of the rule.

Theorem 4.2. *Under Assumption 3.1, for any group $g \in \mathcal{G}$ and $\alpha \in (0,1)$, with probability at least $1-\alpha$,*

$$U_g\left(\hat{t}_g^{\text{mono}}\right) \geq U_g^\star.$$

Proof. The proof of the theorem is inspired by that of Theorem 1 in [3].

For any group g, let $t_g^\star$ be the smallest threshold such that the expected number of relevant items from group g using the threshold policy is larger than or equal to $U_g^\star$, *i.e.*,

$$t_g^\star := \min\left\{t \in \left[t_g^{\max}\right] : U_g(t) \geq U_g^\star\right\}. \quad (5)$$

By Assumption 3.1, we know that the set on the right of Eq.5 is non-empty. Suppose $U_g\left(\hat{t}_g^{\text{mono}}\right) < U_g^\star$, we know $U_g\left(\hat{t}_g^{\text{mono}}\right) < U_g^\star \leq U_g\left(t_g^\star\right)$ by the definition of $t_g^\star$. Thus $\hat{t}_g^{\text{mono}} < t_g^\star$ by the fact that U_g is monotonically increasing. Since $\hat{t}_g^{\text{mono}}$ and $t_g^\star$ are integers, we have $t_g^\star - 1 \geq \hat{t}_g^{\text{mono}}$. By the definitions of $\hat{t}_g^{\text{mono}}$ and $t_g^\star$, this further implies that $t_g^\star > 1$ and

$$\hat{U}_g^-\left(t_g^\star - 1, \alpha\right) \geq U_g^\star > U_g\left(t_g^\star - 1\right).$$

By the lower confidence bound, we know that this happens with probability at most α, which concludes the proof. ∎

Algorithm 1 Fair First-Stage Threshold-Policy Selection

1: **input:** $\mathcal{G}$, $\left(U_g^\star\right)_{g\in\mathcal{G}}$, $\left(t_g^{\max}\right)_{g\in\mathcal{G}}$, $\left(\hat{U}_g^-\right)_{g\in\mathcal{G}}$, α
2: Compute the lower confidence bounds for each group $g \in \mathcal{G}$ and each threshold $t \in \left[t_g^{\max}-1\right]$: $\hat{U}_g^-(t,\alpha)$ for $\hat{t}_g^{\text{mono}}$ or $\hat{U}_g^-\left(t, \frac{\alpha}{t_g^{\max}-1}\right)$ for $\hat{t}_g^{\text{union}}$.
3: Compute $\hat{t}_g$ for each group $g \in \mathcal{G}$: $\hat{t}_g = \hat{t}_g^{\text{mono}}$ by Eq 4 or $\hat{t}_g = \hat{t}_g^{\text{union}}$ by Eq 3.
4: **return** $\pi_{\hat{t}}^f$, where $\hat{t} = \left(\hat{t}_g\right)_{g\in\mathcal{G}}$.

We summarize the fair first-stage threshold-policy selection algorithm in Algorithm 1.

4.2 Finite-Sample Near-Optimality Gaps

So far, we have shown that both threshold selection rules ensure that the expected number of relevant items is large enough with high probability from each group. Now we characterize how far the expected number of relevant items selected by each threshold policy deviates from the target $U_g^\star$ for each group g.

The finite-sample near-optimality gaps we prove depend also on a point-wise upper confidence bound $\hat{U}_g^+(t,\alpha)$ on the expected number of relevant items $U_g(t)$ such that for any group $g \in \mathcal{G}$, threshold $t \in \left[t_g^{\max}-1\right]$, and $\alpha \in (0,1)$,

$$\Pr\left(U_g(t) \leq \hat{U}_g^+(t,\alpha)\right) \geq 1-\alpha. \quad (6)$$

We will show how to derive one such bound for data that takes the form of partial-information user feedback in Section 4.3.

For the union threshold selection rule $\hat{t}_g^{\text{union}}$, we have the following proposition that bounds how much more relevant items it selects than the target $U_g^\star$.

Proposition 4.3. *Under Assumption 3.1, for any group $g \in \mathcal{G}$ and $\alpha \in (0,1)$, with probability at least $1-\alpha$,*

$$U_g\left(\hat{t}_g^{\text{union}}\right) - U_g^\star < \hat{U}_g^+\left(\hat{t}_g^{\text{union}}, \frac{\alpha}{t_g^{\max}-1}\right) - \hat{U}_g^-\left(\hat{t}_g^{\text{union}} - 1, \frac{\alpha}{t_g^{\max}-1}\right).$$

The complete proofs of this and the following propositions are on the arxiv[7]. We can similarly derive the finite-sample near-optimality gap for $\hat{t}_g^{\text{mono}}$ as shown in the following proposition.

Proposition 4.4. *Under Assumption 3.1, for any group g and $\alpha \in (0,1)$, with probability at least $1-\alpha$,*

$$U\left(\hat{t}_g^{\text{mono}}\right) - U_g^\star < \hat{U}_g^+\left(\hat{t}_g^{\text{mono}}, \frac{\alpha}{t_g^{\max}-1}\right) - \hat{U}_g^-\left(\hat{t}_g^{\text{mono}} - 1, \alpha\right).$$

The proof of proposition 4.4 is almost the same as that of Proposition 4.3, and therefore we omit it. We can see that both threshold selection rules rely on the lower confidence bounds, and their optimality gaps depend further on the upper confidence bounds on

[7] https://arxiv.org/abs/2205.15436

the expected number of relevant items for threshold policies. To instantiate the two rules and their finite-sample near-optimality gaps, we introduce how to derive such bounds using user feedback data in the next section.

4.3 Confidence Bounds from User Feedback

In many recommender systems, we have access to an abundance of user feedback (*e.g.*, user clicks, dwell times) that can help reveal the relevance of the items to the queries. However, these data are partial (we only observe a user's feedback for a subset of the items) and biased (by the presentation of the policy that logged the data). Thus, we derive confidence bounds that are robust to these properties[8].

More formally, we use a sample of logged user feedback for m recommendations served by a deployed logging policy π_0. For each recommendation request i, we merely assume that the query and the relevance vector are independently sampled from the query and relevance-vector distribution $q_i, \boldsymbol{r}_i \sim P_{Q,\boldsymbol{R}}$. Instead of directly observing the relevance vector $\boldsymbol{r}_i$, we observe some user feedback (*e.g.*, user clicks) $\boldsymbol{c}_i = \left(c_i^j\right)_{j\in[n]}$ that is typically biased by the recommendation that π_0 made (e.g., position in ranking). To model this presentation bias, we follow the standard approach [1, 2, 22, 35, 67] where the user feedback for an item d^j is decomposed as the product of the user observation and the relevance $c_i^j = o_i^j r_i^j$, where $o_i^j \in \{0,1\}$ denotes whether the user observes the item. The user observation vector $\boldsymbol{o} \in \{0,1\}^n$ is generated from $P^{\pi_0}_{\boldsymbol{O}\mid Q,\boldsymbol{R}}$. We call the conditional probability that the user observes an item under the logging policy π_0 the *propensity*, and denote it as $p_i^j \coloneqq \Pr\left(O_i^j = 1 \mid Q = q_i, \boldsymbol{R} = \boldsymbol{r}_i; \pi_0\right)$. In the case of one-item recommendation, this can be interpreted as the probability that the logging policy π_0 recommends the item. Since we control the logging policy, this propensity is known by design. In the case of ranking, the propensities typically need to be estimated. There are many existing works on estimating the propensities [1, 2, 22, 35, 67] by making assumptions on how the users interact with the ranked items (*e.g.*, position-based click models [1] where the propensity only depends on the rank of the item, and cascade click models [10, 61] where it further depends on the relevances of the items in a particular way). Thus we assume we know the (estimated) propensities, and the batch of logged user feedback for constructing the confidence bounds is $\mathcal{S}_{\mathrm{CB}} = \{q_i, \boldsymbol{c}_i, \boldsymbol{p}_i\}_{i\in[m]}$, where $\boldsymbol{p}_i = \left(p_i^j\right)_{j\in[n]}$ and $m > 1$. Throughout the paper, we assume that the propensities are positive, as formally described below, which can be achieved by carefully designing the logging policy.

Assumption 4.5. *There exists $\gamma > 0$ such that*

$$p^{\sigma^f_{q,g}(j)} > \gamma \quad \forall g \in \mathcal{G}, q \in \mathcal{Q}, j \in \left[t_g^{\max}\right].$$

The confidence bounds we derive using $\mathcal{S}_{\mathrm{CB}}$ are based on the clipped inverse propensity weighted (CIPW) estimator [7, 29, 32] adapted to this ranking setting. The following CIPW estimator estimates the expected number of relevant items $U_g(t_g)$ of threshold policies (recall that we use capital letters to denote random variables)

$$\hat{U}^{\mathrm{CIPW}}_{g,\lambda}\left(t_g\right) \coloneqq \frac{1}{m}\sum_{i\in[m]}\sum_{j\in[t_g]} \min\left(\lambda, \frac{1}{p_i^{\sigma^f_{Q_i,g}(j)}}\right) C_i^{\sigma^f_{Q_i,g}(j)}, \quad (7)$$

with a clipping parameter λ—the maximum inverse propensity weight—to balance the bias and variance of the estimator. The following propositions provide empirical upper and lower confidence bounds on the expected number of relevant items $U_g\left(t_g\right)$ around its CIPW estimate $\hat{U}^{\mathrm{CIPW}}_{g,\lambda}\left(t_g\right)$.

Proposition 4.6. *Under Assumption 4.5, for any $\lambda > 0$, $g \in \mathcal{G}$, $t \in \left[t_g^{\max} - 1\right]$, and $\alpha \in (0,1)$, with probability at least $1-\alpha$, we have that*

$$U_g\left(t\right) \geq \hat{U}^{\mathrm{CIPW}}_{g,\lambda}\left(t\right) - \sqrt{\frac{2V_m\left(\boldsymbol{Z}^{g,t}\right)\ln(2/\alpha)}{m}} - \frac{7t\lambda\ln(2/\alpha)}{3(m-1)} \coloneqq \hat{U}_g^-\left(t,\alpha\right), \quad (8)$$

where $\boldsymbol{Z}^{g,t} = \left(Z_i^{g,t}\right)_{i\in[m]}$ with

$$Z_i^{g,t} = \sum_{j\in[t]} \min\left(\lambda, \frac{1}{p_i^{\sigma^f_{Q_i,g}(j)}}\right) C_i^{\sigma^f_{Q_i,g}(j)}$$

be the CIPW estimate of recommendation request i, and

$$V_m\left(\boldsymbol{Z}^{g,t}\right) = \frac{1}{m(m-1)}\sum_{1\leq i<j\leq m}\left(Z_i^{g,t} - Z_j^{g,t}\right)^2$$

be the sample variance.

Proposition 4.7. *Under Assumption 4.5, for any $\lambda > 0$, $g \in \mathcal{G}$, $t \in \left[t_g^{\max} - 1\right]$, and $\alpha \in (0,1)$, with probability at least $1-\alpha$, we have that*

$$U_g(t) \leq \hat{U}^{\mathrm{CIPW}}_{g,\lambda}(t) + \sqrt{\frac{2V_m\left(\boldsymbol{Z}^{g,t}\right)\ln(4/\alpha)}{m}} + \frac{7t\lambda\ln(4/\alpha)}{3(m-1)}$$
$$+t\sqrt{\frac{\ln(2/\alpha)}{2m}} + \frac{1}{m}\sum_{i\in[m]}\sum_{j\in[t]}\max\left(0, 1-\lambda p_i^{\sigma^f_{Q_i,g}(j)}\right) \coloneqq \hat{U}^+(t,\alpha).$$

Note that both bounds apply to any group of items including the group of all items $\boldsymbol{d}$. Though we apply them for fair first-stage threshold-policy design and characterizing the finite-sample near-optimality gaps, we believe that they might be of independent interests in other contexts.

With these bounds, we can instantiate Algorithm 1 and the finite-sample near-optimality gaps. We can see that the two selection rules can ensure selecting enough relevant items from each group regardless of the accuracy of the relevance model nor the amount of user feedback data across groups. However, the candidate-set sizes of the groups are determined by those factors. In particular, we need to include more items from a group if the relevance model is less accurate and/or we have less data for the group. This shifts the costs of inaccurate relevance estimation and/or lack of data for the items from disadvantaged groups to the latency of the second-stage recommender, and thus provides economic incentives for the decision makers to build accurate relevance models and collect enough data for every group.

[8] The full-information setting is a special case of this partial-information setting where the users observe every item.

4.4 Asymptotic Near-Optimality Analysis

From these upper and lower confidence bounds, we can now analyze how close the selected thresholds are to the optimal thresholds $t_g^\star$ as defined in Eq. 5 asymptotically. In the following propositions, we show that the expected number of relevant items using the selected policies will converge to the target asymptotically, and the selected thresholds will also converge to the optimal thresholds asymptotically, under mild assumptions.

Proposition 4.8. *Let $\hat{t}_g$ be either $\hat{t}_g^{\text{union}}$ or $\hat{t}_g^{\text{mono}}$, and $\lambda = \sqrt{m}$. Under Assumption 3.1 and 4.5, for any group $g \in \mathcal{G}$ and any $\alpha \in (0,1)$, it holds almost surely (with probability 1) that for any $\delta > 0$, there exists $c > 0$ such that for any $m > c$,*

$$U_g^\star - \delta < U_g\left(\hat{t}_g\right) < 1 + U_g^\star + \delta,$$

and even stronger than the second inequality above,

$$U_g\left(\hat{t}_g - 1\right) < U_g^\star + \delta.$$

Proposition 4.9. *Under the conditions in Proposition 4.8 and further assume that U_g is strictly increasing, we have that for any group $g \in \mathcal{G}$, $\alpha \in (0,1)$, it holds almost surely that there exists $c > 0$ such that for any $m > c$,*

$$t_g^\star \le \hat{t}_g \le t_g^\star + 1.$$

5 EMPIRICAL EVALUATION

In this section, we compare the union $\hat{t}_g^{\text{union}}$ and the monotone $\hat{t}_g^{\text{mono}}$ threshold selection ruls with several competitive baselines on first-stage recommendation scenarios simulated from the Microsoft Learning-to-Rank WEB30K dataset [50].

5.1 Experiment Setup

The dataset consists of 30,000 queries, along with the relevances and the features of the items per query. We divide the items into two categories—"old and estabilished" and "new and undiscovered"—by their "url click count" feature given in the dataset. The feature represents "the click count of a url aggregated from user browsing data in a period". We set the items with zero click count as the disadvantaged group *disadv* and the other items as the advantaged group *adv*. We binarize the relevance by assigning relevance 1 to items with an original label of 2, 3, or 4, and 0 to the others. The average numbers of relevant items per query from each group are $\text{AR}_{\text{adv}} = 6.16$ and $\text{AR}_{\text{disadv}} = 13.99$.

For each experiment, we randomly split the data into 1% for training a logistic regression model as the relevance model f, 69% for simulating the user feedback, and 30% for testing different first-stage candidate generation policies. To simulate user feedback, we follow prior works [35] to assume that users follow a position-based click model [16]. More specifically, for each recommendation request i, we randomly sample a query from the 69% data, create a ranking for the top $t_g^{\max}$ items from each group by the relevance model f, simulate the user observation $o_i^j \sim \text{Bernoulli}\left(p_i^j\right)$ with the propensity set as one over the rank of the item $p_i^j = \frac{1}{rank(j \mid \pi_0)}$ for each item j, and the simulated user feedback is $c_i^j = o_i^j r_i^j$.

Unless specified explicitly, we set the size of the user feedback $m = 100,000$, the clipping parameter $\lambda = 100$, and the largest number of items we consider from both groups to be the same $t_{\text{adv}}^{\max} = t_{\text{disadv}}^{\max} = 50$, the success probability $1 - \alpha = 0.9$ by default. We set the target expected numbers of relevant items $U_{\text{adv}}^\star$ and $U_{\text{disadv}}^\star$ to satisfy the equal opportunity constraint [26, 65], *i.e.*, $U_{\text{adv}}^\star / \text{AR}_{\text{adv}} = U_{\text{disadv}}^\star / \text{AR}_{\text{disadv}}$, subject to $U_{\text{disadv}}^\star + U_{\text{adv}}^\star = 5$.

Baselines. We compare $\hat{t}_g^{\text{mono}}$ (*CIPW-LB-mono*) and $\hat{t}_g^{\text{union}}$ (*CIPW-LB-union*) with several baselines. The *Uncalibrated Individual* baseline selects top-ranked items from each group until the sum of the scores predicted by f exceed the target. The *Uncalibrated Marginal* baseline selects the smallest threshold for each group using the simulated user data such that the average sum of scores is greater than the target. The *Platt Individual*, *Platt Marginal*, *Platt PG Individual*, and *Platt PG Marginal* baselines are the same as the uncalibrated ones, except that they apply Platt scaling [49] to the relevance model using user feedback data through inverse propensity weighting [43], where "PG" implies that we calibrate the relevance model per group. The *IPW* baseline selects the smallest threshold such that the inverse propensity weighted (IPW) estimator on the expected number of relevant items of the threshold policy is greater than the target for each group.

Metrics. To compare different first-stage candidate generation policies, we run experiments 50 times for each setting. For each run, we estimate whether each candidate generation policy selects enough relevant items $\text{ER}_g := \mathbb{I}\left\{\hat{\mathbb{E}}\left[\sum_{j\in[n]:d^j\in g} s^j r^j\right] \ge U_g^\star\right\}$ and the candidate-set size $\text{CSS}_g = \hat{\mathbb{E}}_g\left[\sum_{j\in[n]:d^j\in g} s^j\right]$ for both $g = \text{adv}$ and $g = \text{disadv}$ on the 30% full-information test data. We then compare different polices in terms of the percentage of times they ensure selecting enough relevant items (along with standard errors) and the average candidate-set size (along with standard deviations) for both groups across the 50 runs.

5.2 How do different methods scale with the size of user feedback data?

Figure 1 compares different candidate generation policies with respect to the percentage of times they selects enough relevant items ((a) and (b)), and the average candidate-set sizes ((c) and (d)) for both the advantaged groups ((a) and (c)) and the disadvantaged group ((b) and (d)). We can see that the Uncalibrated baselines do not guarantee selecting enough relevant items in general. The IPW baseline selects enough relevant items for most of the instances, and achieves smaller or comparable candidate-set sizes as the two proposed rules. Part of the reason is that we consider discrete threshold policies, and the expected number of relevant items of the optimal threshold $U_g\left(t_g^\star\right)$ is typically larger than the target $U_g^\star$. As long as the IPW estimator does not overestimate the expected number of relevant items more than the difference $U_g\left(t_g^\star\right) - U_g^\star$, it will select enough relevant items. However, as we can see, it fails to select enough relevant items more often on the disadvantaged group, especially when the size of the user feedback data is small. This is because the variance of the CIPW estimator is larger for the disadvantaged group due to a larger threshold and smaller propensities. This phenomenon is more obvious when the relevance model is less accurate for the disadvantaged group as shown in the next subsection. Among the methods that always select enough

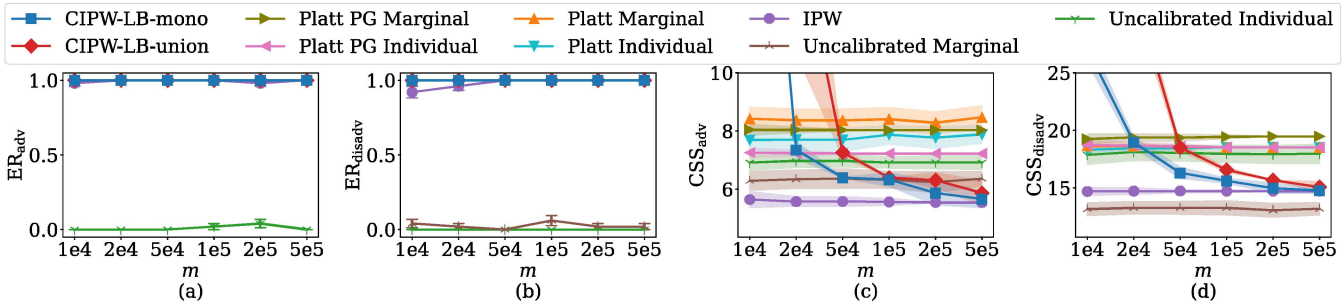

Figure 1: Comparison of different first-stage candidate generation policies when we vary the amount m of user feedback data. The left two plots show the empirical probability, along with standard error bars, that each policy selects enough relevant items for the advantaged group $\mathrm{ER}_{\mathrm{adv}}$ and the disadvantaged group $\mathrm{ER}_{\mathrm{disadv}}$ across 50 runs. The right two plots show the empirical average, along with one standard deviation as shaded regions, of the expected candidate-set size for the advantaged group $\mathrm{CSS}_{\mathrm{adv}}$ and the disadvantaged group $\mathrm{CSS}_{\mathrm{disadv}}$ across 50 runs.

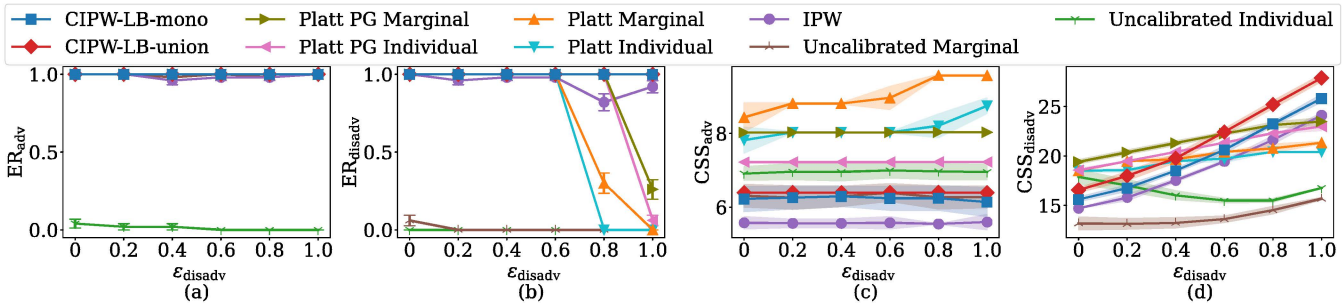

Figure 2: Comparison of different first-stage candidate generation policies when we vary the accuracy of the relevance model to the disadvantaged group by changing the relevance noise ratio $\epsilon_{\mathrm{disadv}}$ to the disadvantaged group.

relevant items, the two proposed threshold selection rules have the smallest candidate-set sizes when the amount of data is large. As we have less and less data, the two proposed rules select more and more items to account for the increasing uncertainty due to the lack of data. Comparing the two proposed rules, the monotone selection rule consistently outperforms the union selection rule in terms of the candidate-set size across data sizes, partly because it allows for a larger failure probability in the lower confidence bounds.

5.3 How does the accuracy of the relevance model affect different groups of items?

To simulate scenarios where the relevance model f might be less accurate for the disadvantaged group, we vary the accuracy of f for the disadvantaged group by replacing its prediction for items in the disadvantaged group with some noise β sampled from $\beta \sim \mathrm{Beta}(1, 10)$, with probability $\epsilon_{\mathrm{disadv}}$, *i.e.*, $f_{\mathrm{disadv}} = (1-\eta) f + \eta\beta$, where $\eta \sim \mathrm{Bernoulli}(\epsilon_{\mathrm{disadv}})$.

Figure 2 compares different candidate generation policies when we vary the relevance noise ratio $\epsilon_{\mathrm{disadv}}$ to the disadvantaged group. We can see that, as the relevance model becomes less accurate for the disadvantaged group ($\epsilon_{\mathrm{disadv}}$ becomes larger), only the two proposed threshold selection rules always select enough relevant items for the disadvantaged group. In particular, IPW fails to select enough relevant items substantially more often for the disadvantaged group than for the advantaged group, since it does not account for the uncertainty in the estimation process. This highlights the benefits of distribution-free and finite-sample guarantees the proposed union and monotone threshold selection rules enjoy. The two rules achieve this by selecting more items from the disadvantaged group to account for the increasing uncertainty due to the less accurate relevance model for the disadvantaged group. Again, the monotone threshold selection rule consistently outperforms the union threshold selection rule in terms of the candidate-set size across different relevance noise ratios to the disadvantaged group.

5.4 How robust are the two proposed rules to the clipping parameter λ?

Figure 3 shows the performance of the two proposed threshold selection rules with varying clipping parameter λ. We can see that the two proposed rules can select enough relevant items across different values in the clipping parameter λ, which confirms their distribution-free and finite-sample guarantees. The candidate-set sizes for both groups exhibit a bowl shape, which is consistent with our theory of finite-sample near-optimality gaps, that the optimal λ lies in the middle where there is a favorable bias and variance trade-off in the CIPW estimator.

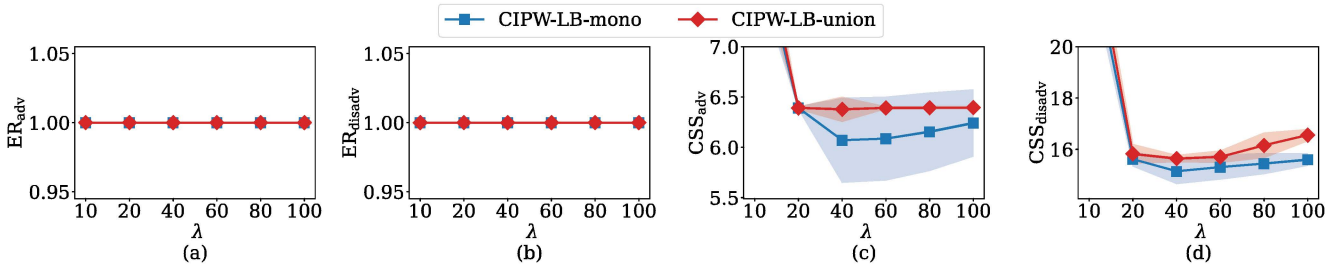

Figure 3: Analysis of the two proposed threshold selection rules when we vary the clipping parameter λ.

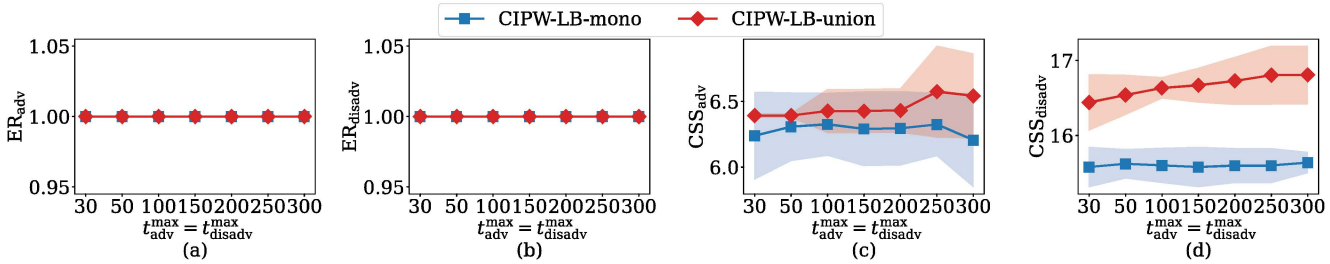

Figure 4: Analysis of the two proposed threshold selection rules when we vary the largest numbers of items we consider for each group, which we set to be the same $t^{\max}_{\mathrm{adv}} = t^{\max}_{\mathrm{disadv}}$.

5.5 Are the two proposed rules robust to the largest considered threshold $t_g^{\max}$?

We compare the performance of the two proposed threshold selection rules when we use different largest considered threshold $t_g^{\max}$ in Figure 4. Unsurprisingly, the two proposed selection rules still select enough relevant items across different values of $t_g^{\max}$, as predicted by the distribution-free and finite-sample guarantees. In terms of the candidate-set size, there is a slight increase for the union threshold selection rule as the largest considered threshold increases. This is expected since it uses smaller and smaller failure probabilities in the lower confidence bounds. For the monotone threshold selection rule, the candidate-set size does not change with the maximum considered threshold, partly because the failure probability it uses in the lower confidence bounds does not change. This also shows that the lower confidence bounds used in the monotone selection rule are still not too loose even when the thresholds are large, since the threshold t is in the log terms in the bounds.

6 CONCLUSION

In this work, we initiated the study of fairness in the first stage of two-stage recommender systems. In particular, we proposed two threshold-policy selection rules that can select fair first-stage policies using abundantly available user-feedback data even if the relevance model used in the first stage is biased and has disparate accuracy across groups. We show that the two selection rules can provably select enough relevant items in expectation from each group with high probability, achieve near-optimal candidate-set sizes, and retain the efficiency of most existing first-stage recommender systems. Both the theoretical analysis and the empirical evaluation confirm that the two proposed selection rules are robust to the amount of user feedback data, the accuracy of the relevance model, and the parameters inside the two rules.

ACKNOWLEDGMENTS

This research was supported in part by NSF Awards IIS-1901168 and IIS-2008139. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *WSDM*.
[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *SIGIR*.
[3] Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael Jordan. 2021. Distribution-free, risk-controlling prediction sets. *JACM* (2021).
[4] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Li Wei, Yi Wu, Lukasz Heldt, Zhe Zhao, Lichan Hong, Ed H Chi, et al. 2019. Fairness in recommendation ranking through pairwise comparisons. In *KDD*.
[5] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*.
[6] Fedor Borisyuk, Krishnaram Kenthapadi, David Stein, and Bo Zhao. 2016. CaSMoS: A framework for learning candidate selection models over structured queries and documents. In *KDD*.
[7] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *JMLR* (2013).
[8] Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. 2018. Balanced neighborhoods for multi-sided fairness in recommendation. In *FAccT*.
[9] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with Fairness Constraints. In *ICALP*.
[10] Praveen Chandar and Ben Carterette. 2018. Estimating clickthrough bias in the cascade model. In *CIKM*.
[11] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. 2019. Top-k off-policy correction for a REINFORCE recommender

system. In *WSDM*.
[12] Yifang Chen, Alex Cuellar, Haipeng Luo, Jignesh Modi, Heramb Nemlekar, and Stefanos Nikolaidis. 2020. Fair contextual multi-armed bandits: Theory and experiments. In *UAI*.
[13] Brian W Collins. 2007. Tackling unconscious bias in hiring practices: The plight of the Rooney rule. *NYUL Rev.* (2007).
[14] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *KDD*.
[15] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *RecSys*.
[16] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *WSDM*.
[17] A Philip Dawid. 1982. The well-calibrated Bayesian. *JASA* (1982).
[18] Fernando Diaz, Bhaskar Mitra, Michael D Ekstrand, Asia J Biega, and Ben Carterette. 2020. Evaluating stochastic rankings with expected exposure. In *CIKM*.
[19] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2021. Two-sided fairness in rankings via Lorenz dominance. In *NeurIPS*.
[20] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *ICML*.
[21] Chantat Eksombatchai, Pranav Jindal, Jerry Zitao Liu, Yuchen Liu, Rahul Sharma, Charles Sugnet, Mark Ulrich, and Jure Leskovec. 2018. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *WWW*.
[22] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention harvesting for context-dependent examination-bias estimation. In *SIGIR*.
[23] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *KDD*.
[24] Stephen Gillen, Christopher Jung, Michael Kearns, and Aaron Roth. 2018. Online learning with an unknown fairness metric. In *NeurIPS*.
[25] Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. 2020. Distribution-free binary classification: prediction sets, confidence intervals and calibration. In *NeurIPS*.
[26] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *NeurIPS*.
[27] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *the Eighth International Workshop on Data Mining for Online Advertising*.
[28] Maria Heuss, Fatemeh Sarvi, and Maarten de Rijke. 2022. Fairness of Exposure in Light of Incomplete Exposure Estimation. In *SIGIR*.
[29] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *JASA* (1952).
[30] Jiri Hron, Karl Krauth, Michael Jordan, and Niki Kilbertus. 2021. On component interactions in two-stage recommender systems. In *NeurIPS*.
[31] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC*.
[32] Edward L Ionides. 2008. Truncated importance sampling. *Journal of Computational and Graphical Statistics* (2008).
[33] Olivier Jeunen and Bart Goethals. 2021. Top-K Contextual Bandits with Equity of Exposure. In *RecSys*.
[34] Thorsten Joachims, Ben London, Yi Su, Adith Swaminathan, and Lequn Wang. 2021. Recommendations as treatments. *AI Magazine* (2021).
[35] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *WSDM*.
[36] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. 2016. Fairness in learning: Classic and contextual bandits. In *NeurIPS*.
[37] Nathan Kallus. 2018. Balanced policy evaluation and learning. In *NeurIPS*.
[38] Wang-Cheng Kang and Julian McAuley. 2019. Candidate generation with binary codes for large-scale top-n recommendation. In *CIKM*.
[39] Matthew Kay, Cynthia Matuszek, and Sean A Munson. 2015. Unequal representation and gender stereotypes in image search results for occupations. In *HCI*.
[40] Till Kletti, Jean-Michel Renders, and Patrick Loiseau. 2022. Introducing the Expohedron for Efficient Pareto-optimal Fairness-Utility Amortizations in Repeated Rankings. In *WSDM*.
[41] Alex Kulesza and Ben Taskar. 2012. Determinantal Point Processes for Machine Learning. *Found. Trends Mach. Learn.* (2012).
[42] Ilja Kuzborskij, Claire Vernade, Andras Gyorgy, and Csaba Szepesvári. 2021. Confident off-policy evaluation and selection through self-normalized importance weighting. In *AISTATS*.
[43] Wonbin Kweon, SeongKu Kang, and Hwanjo Yu. 2021. Obtaining Calibrated Probabilities with Personalized Ranking Models. In *AAAI*.
[44] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. 2020. Off-policy learning in two-stage recommender systems. In *WWW*.
[45] Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. 2018. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *CIKM*.
[46] Harrie Oosterhuis. 2022. Doubly-Robust Estimation for Unbiased Learning-to-Rank from Position-Biased Click Feedback. *arXiv preprint arXiv:2203.17118* (2022).
[47] Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Yadati Narahari. 2020. Achieving Fairness in the Stochastic Multi-Armed Bandit Problem.. In *AAAI*.
[48] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2021. Fairness in rankings and recommendations: an overview. *The VLDB Journal* (2021), 1–28.
[49] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers* (1999).
[50] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR* (2013).
[51] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *ICML*.
[52] Bashir Rastegarpanah, Krishna P Gummadi, and Mark Crovella. 2019. Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In *WSDM*.
[53] Roshni Sahoo, Shengjia Zhao, Alyssa Chen, and Stefano Ermon. 2021. Reliable Decisions with Threshold Calibration. In *NeurIPS*.
[54] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*.
[55] Candice Schumann, Zhi Lang, Nicholas Mattei, and John P Dickerson. 2022. Group fairness in Bandits with Biased Feedback. In *AAMAS*.
[56] Mark Scott. 2017. Google fined record $2.7 billion in EU antitrust ruling. *New York Times* (2017).
[57] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *KDD*.
[58] Yi Su, Lequn Wang, Michele Santacatterina, and Thorsten Joachims. 2019. Cab: Continuous adaptive blending for policy evaluation and learning. In *ICML*.
[59] Adith Swaminathan and Thorsten Joachims. 2015. Batch learning from logged bandit feedback through counterfactual risk minimization. *JMLR* (2015).
[60] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. In *NeurIPS*.
[61] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade model-based propensity estimation for counterfactual learning to rank. In *SIGIR*.
[62] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*. Springer Science & Business Media.
[63] Lequn Wang, Yiwei Bai, Wen Sun, and Thorsten Joachims. 2021. Fairness of exposure in stochastic bandits. In *ICML*.
[64] Lequn Wang and Thorsten Joachims. 2021. User Fairness, Item Fairness, and Diversity for Rankings in Two-Sided Markets. In *ICTIR*.
[65] Lequn Wang, Thorsten Joachims, and Manuel Gomez Rodriguez. 2022. Improving Screening Processes via Calibrated Subset Selection. In *ICML*.
[66] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *SIGIR*.
[67] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *WSDM*.
[68] Yu-Xiang Wang, Alekh Agarwal, and Miroslav Dudık. 2017. Optimal and adaptive off-policy evaluation in contextual bandits. In *ICML*.
[69] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. 2021. Tfrom: A two-sided fairness-aware recommendation model for both customers and providers. In *SIGIR*.
[70] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*.
[71] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. In *NeurIPS*.
[72] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *RecSys*.
[73] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*.
[74] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa* ir: A fair top-k ranking algorithm. In *CIKM*.
[75] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *RecSys*.
[76] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning tree-based deep model for recommender systems. In *KDD*.