Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)
Sister Conferences Best Papers Track

Controlling Fairness and Bias in Dynamic Learning-to-Rank (Extended Abstract) *

Marco Morik[1†], **Ashudeep Singh**[2†], **Jessica Hong**[2] and **Thorsten Joachims**[2]
[1]Technische Universität Berlin
[2]Cornell University
m.morik@tu-berlin.de, ashudeep@cs.cornell.edu, jwh296@cornell.edu, tj@cs.cornell.edu

Abstract

Rankings are the primary interface through which many online platforms match users to items (e.g., news, products, music, video). In these two-sided markets, not only do the users draw utility from the rankings, but the rankings also determine the utility (e.g., exposure, revenue) for the item providers (e.g., publishers, sellers, artists, studios). It has already been noted that myopically optimizing utility to the users – as done by virtually all learning-to-rank algorithms – can be unfair to the item providers. We, therefore, present a learning-to-rank approach for explicitly enforcing merit-based fairness guarantees to groups of items (e.g., articles by the same publisher, tracks by the same artist). In particular, we propose a learning algorithm that ensures notions of amortized group fairness while simultaneously learning the ranking function from implicit feedback data. The algorithm takes the form of a controller that integrates unbiased estimators for both fairness and utility, dynamically adapting both as more data becomes available. In addition to its rigorous theoretical foundation and convergence guarantees, we find empirically that the algorithm is highly practical and robust.

1 Introduction

We consider the problem of dynamic Learning-to-Rank (LTR), where the ranking function dynamically adapts based on the feedback that users provide. Such dynamic LTR problems are ubiquitous in online systems — news-feed rankings that adapt to the number of "likes" an article receives, online stores that adapt to the number of positive reviews for a product, or movie-recommendation systems that adapt to who has watched a movie. In all of these systems, learning and prediction are dynamically intertwined, where past feedback influences future rankings in a specific form of online learning with partial information feedback.

While dynamic LTR systems are in widespread use and unquestionably useful, there are at least two issues that require careful design considerations. First, the ranking system induces a bias through the rankings it presents. In particular, items ranked highly are more likely to collect additional feedback, which in turn can influence future rankings and promote misleading rich-get-richer dynamics [Adamic and Huberman, 2000; Salganik *et al.*, 2006; Joachims *et al.*, 2007; Joachims *et al.*, 2017]. Second, the ranking system is the arbiter of how much exposure each item receives, where exposure directly influences opinion (e.g., the ideological orientation of presented news articles) or economic gain (e.g., revenue from product sales or streaming) for the provider of the item. This raises fairness considerations about how exposure should be allocated based on the merit of the items [Singh and Joachims, 2018; Biega *et al.*, 2018]. We argue that naive dynamic LTR methods that are oblivious to these issues can lead to economic disparity, unfairness, and polarization.

In this paper, we present the first dynamic LTR algorithm – called FairCo – that overcomes rich-get-richer dynamics while enforcing a configurable allocation-of-exposure scheme. Unlike existing fair LTR algorithms [Singh and Joachims, 2019; Biega *et al.*, 2018; Yadav *et al.*, 2019], FairCo explicitly addresses the dynamic nature of the learning problem, where the system is unbiased and fair even though the relevance and the merit of items are still being learned. At the core of our approach lies a merit-based exposure-allocation criterion that is amortized over the learning process. We view the enforcement of this merit-based exposure criterion as a control problem and derive a P-controller that optimizes both the fairness of exposure as well as the quality of the rankings. A crucial component of the controller is the ability to estimate merit (i.e. relevance) accurately, even though the feedback is only revealed incrementally as the system operates, and the feedback is biased by the rankings shown in the process [Joachims *et al.*, 2007]. To this effect, FairCo includes a new unbiased cardinal relevance estimator – as opposed to existing ordinal methods [Joachims *et al.*, 2017; Agarwal *et al.*, 2019] –, which can be used both as an unbiased merit estimator for fairness and as a ranking criterion. In addition to the theoretical justification of FairCo, we provide empirical results on both synthetic news-feed data and real-world movie recommendation data. We find that FairCo is effective at enforcing fairness while providing good ranking performance. Furthermore, FairCo is efficient, robust, and easy to implement.

* Full paper published at ACM SIGIR 2020. See reference [Morik *et al.*, 2020]. † Equal Contribution. Contact Authors.

2 Motivation

Consider the following illustrative example of a dynamic LTR problem. An online news-aggregation platform wants to present a ranking of the top news articles on its front page. Through some external mechanism, it identifies a set $\mathcal{D} = \{d_1, ..., d_{20}\}$ of 20 articles at the beginning of each day, but it is left with the learning problem of how to rank these 20 articles on its front page. As users start coming to the platform, consider a platform that uses the following naive algorithm: maintain a counter $C(d)$ for the number of clicks on each article, and then always sort the articles by the current state of the counts (breaking ties randomly). Unfortunately, this naive algorithm has at least two deficiencies that make it sub-optimal or unsuitable for many ranking applications.

The first deficiency lies in the choice of $C(d)$ as an estimate of average relevance for each article – namely, the fraction of users that want to read the article. Unfortunately, even with infinite amounts of user feedback, the counters $C(d)$ are not consistent estimators of average relevance [Joachims *et al.*, 2007; Joachims *et al.*, 2017]. In particular, items that happened to get more reads in early iterations get ranked highly, where more users find them and thus have the opportunity to provide more positive feedback for them. This perpetuates a rich-get-richer dynamic, where the feedback count $C(d)$ recorded for each article does not reflect how many users actually wanted to read the article.

The second deficiency of the naive algorithm lies in the ranking policy itself, creating a source of unfairness even if the true average relevance of each article was accurately known. Consider the following omniscient variant of the naive algorithm that ranks the articles by their true average relevance (i.e., the true fraction of users who want to read each article). How can this ranking be unfair? Let us assume that we have two groups of articles, G_{right} and G_{left}, with 10 items each (i.e., articles from politically right-leaning and left-leaning sources). 51% of the users (right-leaning) want to read the articles in group G_{right}, but not the articles in group G_{left}. In reverse, the remaining 49% of the users (left-leaning) like only the articles in G_{left}. Ranking articles solely by their true average relevance puts items from G_{right} into positions 1-10 and the items from G_{left} in positions 11-20. This means the platform gives the articles in G_{left} vastly less exposure than those in G_{right}. We argue that this can be considered unfair since the two groups receive disproportionately different outcomes despite having similar merit (i.e., relevance). Here, a 2% difference in average relevance leads to a much larger difference in exposure between the groups.

We argue that these two deficiencies – namely bias and unfairness – are not just undesirable in themselves but that they have undesirable consequences. For example, biased estimates lead to poor ranking quality, and unfairness is likely to alienate the left-leaning users in our example, driving them off the platform and encouraging polarization. Furthermore, note that these two deficiencies are not specific to the news example, but that the naive algorithm leads to analogous problems in many other domains, for example, a ranking system for job applicants or an online marketplace, where rich-get-richer dynamics can encourage gender and race disparity and monopolies, respectively.

These examples illustrate the following two desiderata that a dynamic LTR algorithm should fulfill.

Unbiasedness: The algorithm should not be biased or subject to rich-get-richer dynamics.

Fairness: The algorithm should enforce a fair allocation of exposure based on merit (e.g., relevance).

With these two desiderata in mind, this paper formalizes the dynamic learning-to-rank setup and an amortized notion of merit-based fairness. Finally, we propose a control-based algorithm that is designed to optimize ranking quality while dynamically enforcing fairness.

3 Dynamic Learning-to-Rank

Given is a set of items $\mathcal{D}$ that needs to be ranked in response to incoming requests. At each time step t, a request $\boldsymbol{x}_t, \mathbf{r}_t \sim \mathrm{P}(\boldsymbol{x}, \mathbf{r})$ arrives i.i.d. at the ranking system. Each request consists of a feature vector describing the user's information need $\boldsymbol{x}_t$ (e.g., query, user profile), and the user's vector of true relevance ratings $\mathbf{r}_t$ for all items in the collection $\mathcal{D}$. Only the feature vector $\boldsymbol{x}_t$ is visible to the system, while the true relevance ratings $\mathbf{r}_t$ are hidden. Based on the information in $\boldsymbol{x}_t$, a ranking policy $\pi_t(\boldsymbol{x})$ presents a ranking $\boldsymbol{\sigma}_t$ of items in $\mathcal{D}$ to the user.

After presenting the ranking $\boldsymbol{\sigma}_t$, the system receives a feedback vector $\mathbf{c}_t$ (e.g., clicks) from the user with a non-negative value $\mathbf{c}_t(d) \geq 0$ for every $d \in \mathcal{D}$.

After the feedback $\mathbf{c}_t$ was received, the dynamic LTR algorithm $\mathcal{A}$ now updates the ranking policy and produces the policy π_{t+1} that is used for the next time step.

$$\pi_{t+1} \longleftarrow \mathcal{A}((\boldsymbol{x}_1, \boldsymbol{\sigma}_1, \mathbf{c}_1), ..., (\boldsymbol{x}_t, \boldsymbol{\sigma}_t, \mathbf{c}_t))$$

An instance of such a dynamic LTR algorithm is the naive algorithm outlined in Section 2. It merely computes $\sum \mathbf{c}_t$ to produce a new ranking policy for $t+1$ (here, a global ranking independent of $\boldsymbol{x}$).

Optimizing Ranking Performance. Virtually all ranking metrics used in information retrieval define the utility $U(\boldsymbol{\sigma}|\,\mathbf{r})$ of a ranking $\boldsymbol{\sigma}$ as a function of the relevances of the individual items $\mathbf{r}$. A commonly used utility measure $U(\pi)$ is the DCG [Järvelin and Kekäläinen, 2002] or the NDCG when normalized by the DCG of the optimal ranking. The user-facing goal of dynamic LTR is to converge to the policy $\pi^* = \operatorname{argmax}_\pi U(\pi)$ that maximizes utility. Instead of $\mathbf{c}_t(d)$, if $\mathcal{A}$ had the knowledge of the relevance vector $\mathbf{r}$, the policy that sorts d is optimal for virtually all $U(\boldsymbol{\sigma}|\,\mathbf{r})$ commonly used in IR (e.g., DCG) as also suggested by the Probability Ranking Principle [Robertson, 1977]. So, the problem of finding a ranking that optimizes user utility can be solved by estimating the expected relevance $\mathbf{r}_t(d)$ of each item d conditioned on $\boldsymbol{x}$ using training data comprised of $\mathbf{c}_t$.

3.1 Partial and Biased Feedback

The first key challenge of dynamic LTR lies in the fact that the feedback $\mathbf{c}_t$ provides meaningful feedback only for the items that the user examined. Following a large body of work on click models [Chuklin *et al.*, 2015], we model this as a censoring process using a binary vector $\mathbf{e}_t$ indicating which items were examined by the user, and hence the relationship between $\mathbf{c}_t$ and $\mathbf{r}_t$ is as follows.

$$\mathbf{c}_t(d) = \begin{cases} \mathbf{r}_t(d) & \text{if } \mathbf{e}_t(d) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

A second challenge lies in the fact that the examination vector $\mathbf{e}_t$ cannot be observed. We model the position bias as a probability distribution on the examination vector drawn from a click model as $\mathbf{e}_t \sim \mathrm{P}(\mathbf{e}\,|\boldsymbol{\sigma}_t, \boldsymbol{x}_t, \mathbf{r}_t)$ [Chuklin *et al.*, 2015]. For the simplicity of this paper, we merely use the Position-Based Model (PBM) [Craswell *et al.*, 2008]. It assumes that the marginal probability of examination $\mathbf{p}_t(d)$ for each item d depends only on the rank $\mathrm{rank}(d|\boldsymbol{\sigma})$ of d in the presented ranking $\boldsymbol{\sigma}$.

3.2 Unbiased Estimation of Conditional Relevance

To overcome this problem of unobserved examination vector, we take an approach inspired by [Joachims *et al.*, 2017] and extend it to the dynamic ranking setting. The key idea is to correct for the selection bias with which relevance labels are observed in $\mathbf{c}_t$ using techniques from survey sampling and causal inference [Horvitz and Thompson, 1952]. However, unlike the ordinal estimators proposed in [Joachims *et al.*, 2017], we need cardinal relevance estimates since our fairness disparities are based on cardinal relevance values. We, therefore, propose the following estimator for the regression loss

$$\mathcal{L}^{\mathbf{c}}(w) = \sum_{t=1}^{\tau} \sum_{d} \hat{R}^{\mathrm{w}}(d|\boldsymbol{x}_t)^2 + \frac{\mathbf{c}_t(d)}{\mathbf{p}_t(d)}(\mathbf{c}_t(d) - 2\hat{R}^{\mathrm{w}}(d|\boldsymbol{x}_t)) \quad (2)$$

based on a regressor $\hat{R}^{\mathrm{w}}(d|\boldsymbol{x}_t)$ (e.g., a neural network) with model parameters w. The key idea behind this estimator is that it only uses $\mathbf{c}_t$, but in expectation is equivalent to a least-squares objective that has access to $\mathbf{r}_t$ of the previous τ time steps. This objective corrects for the position bias using Inverse Propensity Score (IPS) weighting [Horvitz and Thompson, 1952; Imbens and Rubin, 2015], where the position bias $(\mathbf{p}_1, ..., \mathbf{p}_\tau)$ takes the role of the missingness model. See the full paper [Morik *et al.*, 2020] for a detailed proof for the unbiasedness property.

Similarly, average relevances can be estimated using the estimator $\hat{R}^{\mathrm{IPS}}(d) = \frac{1}{\tau}\sum_{t=1}^{\tau} \frac{\mathbf{c}_t(d)}{\mathbf{p}_t(d)}$.

4 Fairness in Dynamic LTR

While sorting by unbiased estimates of $R(d|\boldsymbol{x})$ (or $R(d)$ for global rankings) may provide optimal utility to the user, the motivating example in Section 2 illustrates that this ranking can be unfair. There is a growing body of literature to address this unfairness in ranking, and we now extend merit-based fairness [Singh and Joachims, 2018; Biega *et al.*, 2018] to the dynamic LTR setting.

The key scarce resource that a ranking policy allocates among the items is exposure. Based on the position-based model (PBM), exposure of an item d can be defined as the marginal probability of examination $\mathbf{p}_t(d) = \mathrm{P}(\mathbf{e}_t(d) = 1|\,\boldsymbol{\sigma}_t, \boldsymbol{x}_t, \mathbf{r}_t)$. It is the probability that the user will see d and thus have the opportunity to read that article, buy that product, or interview that candidate. For group-based notions of fairness, we aggregate these item-wise examination probabilities into exposure by groups $\mathcal{G} = \{G_1, \ldots, G_m\}$.

$$Exp_t(G_i) = \frac{1}{|G_i|}\sum_{d\in G_i} \mathbf{p}_t(d). \quad (3)$$

These groups can be legally protected groups (e.g., gender, race), reflect some other structure (e.g., items sold by a particular seller), or simply put each item in its own group (i.e., individual fairness).

In order to formulate fairness criteria that relate exposure to merit, we define the merit of an item as its expected average relevance $R(d)$ and again aggregate over groups, i.e, $Merit(G_i) = \frac{1}{|G_i|}\sum_{d\in G_i} R(d)$.

We extend the Disparity of Treatment criterion of [Singh and Joachims, 2018] to the dynamic ranking problem, using an amortized notion of fairness as in [Biega *et al.*, 2018]. In particular, for any two groups G_i and G_j the disparity

$$D_\tau^E(G_i, G_j) = \frac{\frac{1}{\tau}\sum_{t=1}^{\tau} Exp_t(G_i)}{Merit(G_i)} - \frac{\frac{1}{\tau}\sum_{t=1}^{\tau} Exp_t(G_j)}{Merit(G_j)} \quad (4)$$

measures in how far amortized exposure over τ time steps was fulfilled. This **exposure-based fairness disparity** expresses how far, averaged over all time steps, each group of items got exposure proportional to its relevance. We can quantify by how much fairness between all groups is violated using the following overall disparity metric.

$$\overline{D}_\tau = \frac{2}{m(m-1)}\sum_{i=0}^{m}\sum_{j=i+1}^{m}\left|D_\tau(G_i, G_j)\right| \quad (5)$$

Since optimal fairness is achieved for $\overline{D}_\tau = 0$, we seek to minimize $\overline{D}_\tau$. Note that other allocation strategies can be implemented as well by using alternate definitions of disparity [Singh and Joachims, 2018] as discussed in the full paper [Morik *et al.*, 2020].

5 Dynamically Controlling Fairness

Given the formalization of the dynamic LTR problem, our definition of fairness, and our derivation of estimators for all relevant parameters, we formulate the following control problem that is robust to the uncertainty in the estimates $\hat{R}(d|\boldsymbol{x})$ and $\hat{R}(d)$ at the beginning of the learning process. Our method, called *FairCo*, takes the form of a Proportional Controller (a.k.a. P-Controller) [Bequette, 2003]. P-controllers are a widely used control-loop mechanism that applies feedback through a correction term that is proportional to the error. In our application, the error corresponds to the violation of our amortized fairness disparity from Equation (4). Specifically, for any set of disjoint groups $\mathcal{G} = \{G_1, \ldots, G_m\}$, the error term of the controller for any item d is defined as

$$\forall G \in \mathcal{G}\ \forall d \in G : \mathbf{err}_\tau(d) = (\tau - 1)\cdot \max_{G_i}\left(\hat{D}_{\tau-1}(G_i, G)\right).$$

The error term $\mathbf{err}_\tau(G)$ is zero for the group that already has the maximum exposure w.r.t. its merit. For items in the other

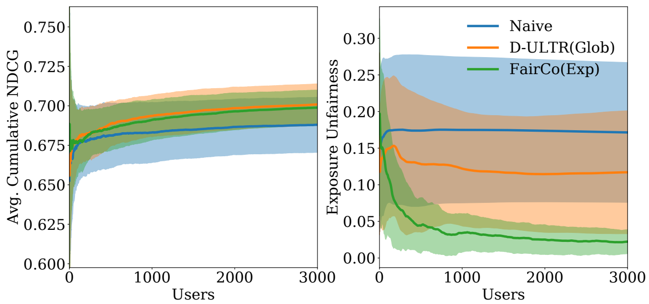

Figure 1: Convergence of NDCG (left) and Unfairness (right) as the number of users increases.

groups, the error term grows with increasing disparity. Using this error term, we can state the FairCo ranking policy as

$$\text{FairCo:} \qquad \boldsymbol{\sigma}_\tau = \underset{d \in \mathcal{D}}{\operatorname{argsort}} \left(\hat{R}(d|\boldsymbol{x}) + \lambda \, \mathbf{err}_\tau(d) \right). \tag{6}$$

For the exposure-based disparity $\hat{D}^{E}_{\tau-1}(G_i, G)$, we refer to this policy as FairCo(Exp), and note that the disparity $\hat{D}_{\tau-1}(G_i, G)$ in the error term uses the estimated $\hat{Merit}(G)$ from Equation (3.2), which converges to $Merit(G)$ as the τ increases.

Similar to the naive ranking policy, FairCo is a sort-based policy. However, the sorting criterion is a combination of relevance $\hat{R}(d|\boldsymbol{x})$ and an error term representing the fairness violation. The idea behind FairCo is that the error term pushes the items from the underexposed groups upwards in the ranking. The parameter λ can be chosen to be any positive constant, but a suitable choice of λ can have an influence on the finite-sample behavior of FairCo: a higher λ can lead to an oscillating behavior, while a smaller λ makes the convergence smoother but slower. We explore the role of λ in the experiments but find that keeping it fixed at $\lambda = 0.01$ works well across all of our experiments. Another key quality of FairCo is that it is agnostic to the choice of error metric, and we conjecture that it can easily be adapted to other types of fairness disparities.

6 Experimental Evaluation

We conducted an empirical evaluation on a semi-synthetic news dataset to investigate different aspects of the proposed methods under controlled conditions. In the news dataset, each news article either belongs to group G_{left} or G_{right} (same as the example in Section 2) and the users have a polarity score specifying their preferences. The users sequentially arrive to the system and provide clicks which we use to measure NDCG (user utility) and the amortized unfairness under the following ranking policies.

Naive: Rank by the sum of the observed feedback $\mathbf{c}_t$.

D-ULTR(Glob): Rank by unbiased $\hat{R}^{\text{IPS}}(d)$ from Eq. (3.2).

FairCo(Exp): Fairness controller from Eq. (6).

6.1 Can FairCo Reduce Unfairness While Maintaining Good Ranking Quality?

This is the key question in evaluating FairCo, and Figure 1 shows how NDCG and Unfairness converge for Naive, D-ULTR(Glob), and FairCo(Exp). The plots show that Naive

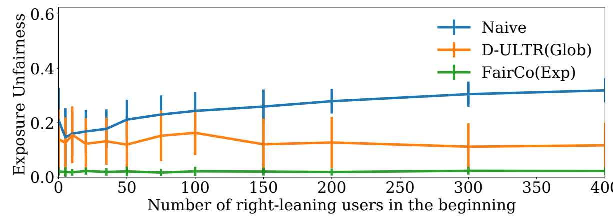

Figure 2: The effect of an initial block of right-leaning users on the Unfairness of Exposure.

achieves the lowest NDCG and that its unfairness remains high as the number of user interactions increases. D-ULTR(Glob) achieves the best NDCG, as predicted by the theory, but its unfairness is only marginally better than that of Naive. Only FairCo manages to substantially reduce unfairness, and this comes only at a small decrease in NDCG compared to D-ULTR(Glob).

6.2 Does FairCo Overcome the Rich-Get-Richer Dynamic?

We argue that naively ranking items is highly sensitive to the initial conditions (e.g. which items get the first clicks), leading to a rich-get-richer dynamic. The experiment shown in Figure 2 tests whether FairCo overcomes this problem. In particular, we adversarially modify the user distribution so that the first x users are right-leaning, followed by x left-leaning users, before we continue with a balanced user distribution, and measure the unfairness after 3000 user interactions. As expected, Naive is the most sensitive to the head-start that the right-leaning articles get. D-ULTR(Glob) fares better and its unfairness remains constant but independent of the initial user distribution because the unbiased estimator $\hat{R}^{\text{IPS}}(d)$ corrects for the presentation bias so that the estimates still converge to the true relevances. FairCo inherits this robustness to initial conditions since it uses the same $\hat{R}^{\text{IPS}}(d)$ estimator, and its active control for unfairness makes it achieve low unfairness across the whole range.

7 Conclusions

In this work, we identify how biased feedback and uncontrolled exposure allocation can lead to unfairness and undesirable behavior in dynamic LTR. We propose FairCo to address this problem, which adaptively enforces amortized merit-based fairness constraints while the underlying relevances are still being learned. The algorithm is robust to presentation bias and thus does not exhibit rich-get-richer dynamics. FairCo is easy to implement and computationally efficient making it well suited for practical applications.

Acknowledgements

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

References

[Adamic and Huberman, 2000] Lada A Adamic and Bernardo A Huberman. Power-law distribution of the world wide web. *Science*, 2000.

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

[Bequette, 2003] B Wayne Bequette. *Process control: modeling, design, and simulation*. Prentice Hall Professional, 2003.

[Biega *et al.*, 2018] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *SIGIR*, 2018.

[Chuklin *et al.*, 2015] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 2015.

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

[Horvitz and Thompson, 1952] Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 1952.

[Imbens and Rubin, 2015] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[Järvelin and Kekäläinen, 2002] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *TOIS*, 2002.

[Joachims *et al.*, 2007] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM TOIS*, 2007.

[Joachims *et al.*, 2017] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. Unbiased learning-to-rank with biased feedback. In *WSDM*, 2017.

[Morik *et al.*, 2020] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*, pages 429–438, 2020.

[Robertson, 1977] Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 1977.

[Salganik *et al.*, 2006] Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 2006.

[Singh and Joachims, 2018] Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *ACM SIGKDD*, 2018.

[Singh and Joachims, 2019] Ashudeep Singh and Thorsten Joachims. Policy learning for fairness in ranking. In *NeurIPS*. 2019.

[Yadav *et al.*, 2019] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. Fair learning-to-rank from implicit feedback. *arXiv preprint arXiv:1911.08054*, 2019.