

# Maximizing Marginal Fairness for Dynamic Learning to Rank

Tao Yang  
University of Utah  
Salt Lake City, Utah  
taoyang@cs.utah.edu

Qingyao Ai  
University of Utah  
Salt Lake City, Utah  
aiqy@cs.utah.edu

## ABSTRACT

Rankings, especially those in search and recommendation systems, often determine how people access information and how information is exposed to people. Therefore, how to balance the relevance and fairness of information exposure is considered as one of the key problems for modern IR systems. As conventional ranking frameworks that myopically sorts documents with their relevance will inevitably introduce unfair result exposure, recent studies on ranking fairness mostly focus on dynamic ranking paradigms where result rankings can be adapted in real-time to support fairness in groups (i.e., races, genders, etc.). Existing studies on fairness in dynamic learning to rank, however, often achieve the overall fairness of document exposure in ranked lists by significantly sacrificing the performance of result relevance and fairness on the top results. To address this problem, we propose a fair and unbiased ranking method named Maximal Marginal Fairness (MMF). The algorithm integrates unbiased estimators for both relevance and merit-based fairness while providing an explicit controller that balances the selection of documents to maximize the marginal relevance and fairness in top-k results. Theoretical and empirical analysis shows that, with small compromises on long list fairness, our method achieves superior efficiency and effectiveness comparing to the state-of-the-art algorithms in both relevance and fairness for top-k rankings.

## CCS CONCEPTS

• Information systems → Learning to rank.

## KEYWORDS

Learning to Rank, Ranking Fairness, Unbiased Learning

### ACM Reference Format:

Tao Yang and Qingyao Ai. 2021. Maximizing Marginal Fairness for Dynamic Learning to Rank. In *Proceedings of the Web Conference 2021 (WWW '21)*, April 19–23, 2021, Ljubljana, Slovenia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3442381.3449901>

## 1 INTRODUCTION

Fairness in ranking has drawn much attention as ranking systems, especially those in search and recommendation systems, could significantly affect how people access information and how information is exposed to users [5]. For example, job posts ranked highly

on LinkedIn are more likely to receive more applications; new products on the bottom of Amazon search result pages are less likely to be clicked by customers. Without proper treatments, ranking systems could introduce unintended unfairness to various aspects of people's lives, such as job opportunities, economical gain, etc.

As traditional ranking algorithms that produce static ranked lists can hardly handle the unfairness of result exposure in practice, the ranking paradigms that dynamically change result ranking on the fly, namely the dynamic Learning-to-Rank (LTR) [20], have received more and more attention in the research communities. The idea of dynamic LTR is to learn and adapt ranking models based on user feedback in real time so that past user interactions or result distributions could affect the exposure of future results. Through heavily exposed to implicit user examination bias [14, 16, 33] and selection bias [21, 28], dynamic LTR allows ranking systems to produce multiple ranked lists for a single request, which makes it possible to explicitly control or balance the exposure of results in different groups (i.e., race, gender, etc.) for ranking fairness.

Existing studies on ranking fairness in dynamic LTR mostly focuses on developing effective algorithms to achieve merit-based fairness [5, 25]. Well-known examples include the linear programming algorithm (LinProg) [25] that determines result ranking by taking group fairness as an optimization constraint, and the FairCo algorithm [20] that manipulates ranking scores dynamically according to the current exposure of results in different groups. Despite their solid theoretical foundations, these algorithms often sacrifice the performance of result relevance significantly on the top results of each ranked list. These trade-offs are uncontrollable as they are difficult to be quantified explicitly, which makes the risk of applying fairness algorithms in practice still high today.

In this paper, we present a novel fairness algorithm for dynamic LTR that considers both relevance and fairness for ranking criterion. We believe that it is important to control and balance result relevance and fairness in online ranking systems, especially for the top-k results in ranked lists. Therefore, inspired by the studies of search diversification that try to improve novelty while preserving ranking performance (especially the maximal marginal relevance frameworks [6]), we propose a Maximal Marginal Fairness (MMF) algorithm that optimizes ranking performance while explicitly mitigating amortized group unfairness for selected items. MMF dynamically selects documents that are relevant and underexposed in top-k results to maximize the marginal relevance and fairness. With a small compromise on the bottom of long ranked lists, MMF achieves superior performance and significantly outperforms the state-of-the-art fairness algorithms not only in top-k relevance, but also in top-k fairness. As most people would examine or only examine the top results on a result page [8, 13, 19, 27], MMF is highly competitive and preferable in real LTR systems and applications.

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-8312-7/21/04.

<https://doi.org/10.1145/3442381.3449901>

From technical viewpoints, the main contribution of this paper are two-fold. First, extending the existing definition of merit-based group fairness in ranking [5, 25], we develop a metric to measure the group fairness of exposure in the top-k rankings. We show that most existing state-of-the-art methods for ranking fairness focus more on the overall fairness of document exposure while compromising a lot in the top ranks of each ranked list. Second, we propose a Maximal Marginal Fairness (MMF) algorithm that can explicitly control and balance result relevance and fairness in top-k rankings. In particular, our method uses a hyper-parameter  $\lambda$  to determine whether the system needs to select an item being relevant or an item that improves the marginal fairness of the current ranked list. We evaluate and compare our algorithm with existing state-of-the-art methods on both synthetic and real-world preference datasets using simulated user interactions. Theoretical and empirical analysis shows that our method can achieve significant efficiency and effectiveness improvements in top-k relevance and fairness.

## 2 RELATED WORK

Leveraging biased click data for optimizing learning to rank systems has been a popular approach in information retrieval [13, 14]. As click data are often noisy and biased, numerous unbiased learning to rank (ULTR) methods have been proposed based on different theoretical foundations [4], including click models [9, 12, 29], randomization [22], causal inference [1, 3, 17], etc. Those methods make it possible to achieve unbiased relevance estimation or rankings for learning to rank in noisy environments.

Ranking according to intrinsic result relevance is important yet not enough today. One of the key principle [23] of ranking in Information Retrieval states that documents should be ranked in order of the probability of relevance or usefulness. Such arguments, however, only consider the responsibilities of ranking systems to users while ignoring the items being ranked. Recently, there has been a growing concern about fairness behind ranking algorithms in both academia and industry [5, 34, 35]. Since rankings are the main interface from which we find content, products, music, news and etc., their ordering contributes not only to the utility of users, but also to information providers.

To address this, some studies focus on restricting the fraction of items of each attribute in a ranking [32, 34]. While another natural way of understanding unfairness is by considering difference in exposure, which directly relates unfairness to economic or social opportunities [35]. For example, researchers try to achieve amortized fairness of exposure by reducing group disparate exposure for only the top position [35] or making exposure proportional to relevance [5, 20, 25, 26]. The reason for reducing exposure disparity is that minimal difference in relevance can result in a large difference in exposure across groups [25] if the ranking only is based on relevance, since there might exist a large skew in the distribution of exposure like position bias [15], i.e., users observe way much more top ranks than the bottom part. To achieve this, various methods have been proposed. For example, fairness loss based on Softmax function is proposed by Zehlike and Castillo [35], which considers equal group exposure in top rank. Linear programming methods are proposed in [5, 7, 25], which try to give a general framework for computing optimal probabilistic rankings for merit-based fairness.

**Table 1: A summary of notations.**

$\sigma_t, \mathbf{x}_t, \mathbf{c}_t, \mathbf{o}_t, \mathbf{p}_t, \mathbf{r}_t$	The presented document ranking ( $\sigma_t$ ), the corresponding feature vectors ( $\mathbf{x}_t$ ), the clicks on each document ( $\mathbf{c}_t$ ), the binary variables indicating user’s examination on each document ( $\mathbf{o}_t$ ), user’s propensity to examine results on certain positions ( $\mathbf{p}_t$ , namely $P(\mathbf{o}_t = 1)$ ), and the true (personalized) binary relevance rating of the documents ( $\mathbf{r}_t$ ) at time step $t$ in dynamic LTR setting.
$R(d)$	The average relevance across all users for document $d$ .
$R_\theta(d \mathbf{x}_t)$	The predicted relevance of $d$ given by the model parameterized by $\theta$ .
$\mathcal{G}, G_i, \mathbb{G}_i^k$	The set of groups to consider ( $\mathcal{G}$ ) and the $i$ th group ( $G_i$ ). $\mathbb{G}_i^k$ is the priority queue of size $k$ for $G_i$ according to estimated relevance.

The policy learning method is adopted to maximize ranking metrics while introducing unfairness as additional loss in [26].

In fact, the ideas behind existing fairness methods are always to avoid showing items from the same class or group, which is not a new topic and have already been studied as diversity and novelty in the information Retrieval Community for decades [11, 30, 31]. Though the utility of diversified ranking is still centered on users not for providers, studies on search diversification also focus on optimizing result ranking beyond information relevance. One of the most famous approaches is the maximal marginal relevance approach (MMR) [6] that treats ranking as a Markov Decision Process by selecting documents that maximize the combination of sub-topic relevance given previously selected results.

In a recent work [20], the proportional controller in control theory is applied to mitigate unfairness in dynamic LTR settings. However, existing fairness algorithms in dynamic LTR only consider overall fairness while ignoring the top-k unfairness. In this paper, we first define the top-k group fairness, which is often ignored by existing works. Then, based on the top-k fairness metrics, inspired by MMR, we propose a concept named marginal fairness and directly optimize top-k relevance and fairness in a dynamic LTR environment where both the relevance and fairness modules of the algorithms are learned and adapted according to real-time user feedback.

Another algorithm is related with our work is FA\*IR, proposed by Zehlike et al. [34]. FA\*IR is a post-processing method to explicitly guarantee the exposure of documents in top-k ranking. The key difference between our algorithm MMF and FA\*IR is that FA\*IR focuses on a simplified offline scenario and ignore the fact that result exposure could affect relevance estimations in LTR.

## 3 PROBLEM FORMULATION

In this section, we introduce the problem of relevance and fairness estimation in dynamic LTR with a focus on the top-k results. A summary of the notations used in this paper is shown in Table 1.

### 3.1 Relevance Estimation in Dynamic LTR

In dynamic LTR frameworks, the most popular paradigm for relevance estimation is to infer relevance from user feedback directly. Specifically, from partial and biased feedback such as user clicks, we need to construct a model to estimate relevance. In general, traditional LTR models can be categorized as cardinal and ordinal ones. Ordinal LTR models give ordinal numbers according to output scores for items, while those scores themselves have no exact meaning. Cardinal LTR models, on the other hand, predict document ranking scores that are proportional or directly reflect

the relevance of the documents. Since exposure disparity explicitly involves relevance, similar to previous studies [20], we only focus on cardinal LTR models in this paper. Here, we adopt a LTR model  $R_\theta(d|\mathbf{x}_t)$  parameterized by  $\theta$  with least-square loss as

$$\begin{aligned} \mathcal{L}^\tau(\theta) &= \sum_{t=1}^{\tau} \sum_d \left( \mathbf{r}_t(d) - R_\theta(d|\mathbf{x}_t) \right)^2 \\ &\triangleq \sum_{t=1}^{\tau} \sum_d \left( R_\theta(d|\mathbf{x}_t)^2 - 2\mathbf{r}_t(d)R_\theta(d|\mathbf{x}_t) \right) \end{aligned} \quad (1)$$

where  $\tau$  is the total number of existing time steps in dynamic LTR, and  $\triangleq$  means equal while ignoring constants.

As the true relevance judgements  $\mathbf{r}_t$  are not available, based on the studies of unbiased learning to rank [17], we define an unbiased estimation of  $\mathcal{L}^\tau(\theta)$  using click data  $\mathbf{c}_t$  by applying Inverse Propensity Score (IPS) weighting as

$$\tilde{\mathcal{L}}^\tau(\theta) = \sum_{t=1}^{\tau} \sum_d \left( R_\theta(d|\mathbf{x}_t)^2 - 2 \frac{\mathbf{c}_t(d)}{\mathbf{p}_t(d)} R_\theta(d|\mathbf{x}_t) \right) \quad (2)$$

where  $\mathbf{p}_t$  is the user's examination propensities on each result position (namely  $P(\mathbf{o}_t = 1)$ ), and we assume that

$$P(c = 1) = P(o = 1) \cdot P(r = 1) \quad (3)$$

which means that users click a search result ( $c = 1$ ) only when it is both observed ( $o = 1$ ) and perceived as relevant ( $r = 1$ ), and  $o$  and  $r$  are independent. The unbiasedness of  $\tilde{\mathcal{L}}^\tau(\theta)$  can be proven as below

$$\begin{aligned} \mathbb{E}_{\mathbf{o}_t}[\tilde{\mathcal{L}}^\tau(\theta)] &= \sum_{t=1}^{\tau} \sum_d \left( R_\theta(d|\mathbf{x}_t)^2 - 2 \frac{\mathbb{E}_{\mathbf{o}_t}[\mathbf{c}_t(d)]}{\mathbf{p}_t(d)} R_\theta(d|\mathbf{x}_t) \right) \\ &= \sum_{t=1}^{\tau} \sum_d \left( R_\theta(d|\mathbf{x}_t)^2 - 2 \frac{\mathbf{p}_t(d)\mathbf{r}_t(d)}{\mathbf{p}_t(d)} R_\theta(d|\mathbf{x}_t) \right) \\ &= \sum_{t=1}^{\tau} \sum_d \left( R_\theta(d|\mathbf{x}_t)^2 - 2\mathbf{r}_t(d)R_\theta(d|\mathbf{x}_t) \right) = \mathcal{L}^\tau(\theta) \end{aligned} \quad (4)$$

Similarly, we could get unbiased estimation of the average relevance of  $d$  across all users ( $R(d)$ ) as

$$R^{IPS}(d) = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{\mathbf{c}_t(d)}{\mathbf{p}_t(d)} \quad (5)$$

$R^{IPS}(d)$  is used for both fairness controlling and as a global ranking baseline without personalization (i.e., considering document relevance with respect to each individual user). The estimation of position bias  $\mathbf{p}_t(d)$  can be achieved through various methods in advance [2, 3, 29], which is not in the scope of this paper.

### 3.2 Merit-based Top-k Fairness

We now introduce the definition of the Top-k merit-based group unfairness in dynamic LTR. First, similar to previous studies [5, 20, 25], we define the exposure of a document  $d$  as its marginal probability of being examined  $\mathbf{p}_t(d) = P(\mathbf{o}_t(d) = 1|\sigma_t, \mathbf{x}_t, \mathbf{r}_t)$  where  $\sigma_t, \mathbf{x}_t, \mathbf{r}_t$  are the presented ranking, feature vectors, and true relevance of documents at time step  $t$  in dynamic LTR. Let  $\mathcal{G} = \{G_1, \dots, G_m\}$  be the possible groups that each document could belong to. Suppose that we only care about the exposure fairness in

top-k positions, then we can define group-based fairness in top-k positions following the merit-based fairness definition [20] as

$$Exp_t^k(G_i) = \frac{1}{|G_i|} \sum_{d \in G_i \cap d \in \sigma_t^k} p_t(d) \quad (6)$$

where  $\sigma_t^k$  is the top k documents in presented ranking  $\sigma_t$ ,  $\mathbf{p}_t(d)$  is the examination propensity on  $d$  at time step  $t$ . Following previous studies, we define the merit of a group or document as the expected average relevance across all documents in the same group:

$$Merit(G_i) = \frac{1}{|G_i|} \sum_{d \in G_i} R(d) \quad (7)$$

Thus, for any two groups  $G_i$  and  $G_j$ , we can define their accumulative disparity in top-k positions as

$$\begin{aligned} Exp\_cum_\tau^k(G_i) &= \sum_{t=1}^{\tau} Exp_t^k(G_i) \\ Exp\_Mer_\tau^k(G_i) &= \frac{\frac{1}{\tau} Exp\_cum_\tau^k(G_i)}{Merit(G_i)} \end{aligned} \quad (8)$$

$$D_\tau^k(G_i, G_j) = \left\| Exp\_Mer_\tau^k(G_i) - Exp\_Mer_\tau^k(G_j) \right\|$$

Thus, larger disparity indicates greater violation of fairness in top-k rankings. Additional to pairwise disparity defined in Eq.8, for ranking with more than 2 groups, we define the unfairness of a list as the average disparity of all pairs:

$$Unfairness@k = \overline{D_\tau^k} = \frac{2}{m(m-1)} \sum_{i=0}^m \sum_{j=i+1}^m \left\| D_\tau^k(G_i, G_j) \right\| \quad (9)$$

Note that the original merit-based unfairness defined in [20] can be seen as a special case in our formulation where  $k$  is the number of all possible documents. In this paper, our goal is to create rankings with high relevance  $\mathbf{r}_\tau$  while maintaining a low unfairness  $\overline{D_\tau^k}$  in top-k results after the time step  $\tau$  in dynamic LTR.

## 4 OUR APPROACH

In this section, we describe our approach to balance relevance and fairness in dynamic LTR. Specifically, we first introduce the concept of maximal marginal fairness, then we discuss our MMF algorithm.

### 4.1 Maximal Marginal Fairness

In this paper, we define a new concept for ranking fairness as *Marginal Fairness*. Ranking can be modeled as a greedy selection process where we create a ranked list by sequentially selecting documents from a candidate pool. Based on this assumption, considerable ranking algorithms have been proposed to optimize ranking utility from various perspectives such as information relevance [18], novelty [24], etc. Particularly, in the studies of search diversification, one of the most well-known algorithms is the Maximal Marginal Relevance (MMR) algorithm [6] that greedily selects documents based on their *Marginal Relevance*, the maximal utility of a document given the selected results in the current ranked list, to balance ranking relevance and novelty. Inspired by MMR and the concept of marginal relevance, we model ranking as a greedy selection problem and define marginal fairness as *the marginal gain of fairness, or*

the marginal reduction of unfairness, when selecting and adding a document given the selected results in the current ranked list.

$d_\tau^k$  is the document we select for  $k^{th}$  position in the ranked list  $\sigma_\tau$ . Formally, then the marginal fairness of selecting document  $d_\tau^k$  from group  $G$  is

$$MF(G|\sigma_\tau^{k-1}) = \overline{D_\tau^{k-1}} - \overline{D_\tau^k}(d_\tau^k), \text{ where } d_\tau^k \in G \quad (10)$$

where  $\overline{D_\tau^k}(d_\tau^k)$  is the unfairness after we select  $d_\tau^k$ .

Then, to maximize merit-based fairness of top  $k$  results, a straightforward method is to maximize marginal fairness by selecting the document with maximum  $MF(G|\sigma_\tau^{k-1})$ . This observation serves as our foundation for the construction of the MMF algorithm.

## 4.2 Algorithm

The MMF algorithm includes three sub-modules – the selection of documents for maximizing marginal fairness, the selection of documents for maximizing relevance, and the controller that balances top-k relevance and fairness.

**4.2.1 Fairness Module.** As discussed previously, MMF optimizes top-k fairness through greedily selecting documents to maximize marginal fairness. In Eq. (10), the computation of marginal fairness requires us to compute the current and updated disparities ( $D_\tau^{k-1}(G_i, G_j)$  and  $D_\tau^k(G_i, G_j)$ ) for every pair of groups. In practice, however, we can prove that maximal marginal fairness can be achieved directly by selecting a document from the group with lowest expected merit  $Exp\_Mer_\tau^k(G_i)$  in Eq. (8) given that the scale of  $\mathbf{p}_\tau(d)$  is much smaller than  $Exp\_Mer_\tau^k(G_i)^1$ . Also, as the true relevance of documents  $R(d)$  is not available, we use  $R^{IPS}(d)$  to estimate  $R(d)$  and get an unbiased estimation of  $Exp\_Mer_\tau^k(G_i)$  as  $Exp\_Mer_\tau^k(G_i)$ .

Thus, we compute the best group to select documents from MMF as

$$G_\tau^k = \max_G MF(G|\sigma_\tau^{k-1}) = \min_G Exp\_Mer_\tau^k(G) \quad (11)$$

Note that every document in  $G_\tau^k$  would have the maximal marginal fairness given the ranked list  $\sigma_\tau^{k-1}$ .

**4.2.2 Relevance Module.** To maximize the performance of top-k results in terms of ranking relevance, the optimal solution is to select documents based on their estimated relevance from user interactions (i.e.,  $R^{IPS}$ ) or a LTR model parameterized by  $\theta$  ( $R^\theta$ ) as

$$\tilde{d}_\tau^k = \operatorname{argmax}_{d \notin \sigma_\tau^{k-1}} R^\theta(d) \quad (12)$$

where we only consider relevance but not fairness in selecting the next document for the ranked list. In case we need to select documents from the group with maximal marginal fairness, we could apply similar process and get

$$\tilde{d}_\tau^k = \operatorname{argmax}_{d \in G_\tau^k \wedge d \notin \sigma_\tau^{k-1}} R^\theta(d) \quad (13)$$

In practice, Eq. (12) and Eq. (13) can be computed together by maintaining multiple priority queues  $\mathbb{G}$ , for each group separately.

<sup>1</sup>We ignore the proof for simplicity

**4.2.3 Balance between Relevance and Fairness.** MMF implements a simple yet effective method to control the relevance and fairness of top-k rankings in dynamic LTR by adding a stochastic controller parameterized with  $\lambda$ . Intuitively, the idea of the controller is to choose the final selected item  $\tilde{d}_\tau^k$  between the document that maximizes ranking relevance  $\tilde{d}_\tau^k$  and the document maximizing marginal fairness  $\tilde{d}_\tau^k$  with probability  $\lambda$ . Formally, we have

$$\tilde{d}_\tau^k \sim (\lambda \tilde{d}_\tau^k + (1 - \lambda) \tilde{d}_\tau^k) \quad (14)$$

The greater  $\lambda$  is, the fairer the ranking is and the less ranking relevance is. Different from the linear trade-off strategy used in other fairness algorithms [20, 25] that directly combines the fairness and relevance scores before the document selection process, we adopt a probabilistic strategy to strike the balance, which not only makes the algorithm more robust to the magnitude of estimated relevance and fairness scores, but also provides explicit functions to balance relevance and fairness in practical applications.

The overall MMF is shown in Algorithm 1. At step 1, we do initialization. At step 2, a user enters the dynamic LTR system. At step 3 – 7, we dynamically learn a LTR model  $R^\theta$  or directly use the averaged inverse propensity weighted clicks  $R^{IPS}$  to estimate the relevance of items. The LTR model has advantages over  $R^{IPS}$  as it can estimate personalized relevance if features contain user information. At step 8 – 10, we construct priority queues for each group with estimated relevance from step 3 – 7. In practice, we could construct priority queues while estimating relevance at the time. At step 12 – 21, we select the item according to Equations (12) to (14). At step 15,  $G_{ind}$  is a function to get group index of an item. At step 22, we collect clicks in a real-world application or sample clicks according to Eq.3 for our experiments. At step 23 – 27, we update the relevance estimation accordingly. Note that, for different time steps, users interact with the same item candidates at step 2 – 27.

## 4.3 Complexity Analysis

To illustrate the efficiency of MMF, we conduct complexity analysis and use the state-of-the-art fairness algorithm, i.e., FairCo [20]. FairCo is one of the most efficient fairness algorithms for dynamic LTR. It achieves fairness by dynamically adding perturbations to the ranking scores of documents based on the exposure of different groups. As relevance is estimated separately, we only discuss the complexity of fairness controlling in FairCo and MMF.

**4.3.1 Time Complexity.** Let  $k$  be the number of ranks we care about,  $|\mathcal{G}|$  be the number of groups, and  $n$  be the total number of documents to rank. As FairCo needs to add score perturbations to all documents, it need to track the cumulative group exposure of all documents (i.e.,  $Exp\_cum_\tau^k(G_i)$  in Eq. (8)) and do linear interpolation with  $O(n)$  time. It needs  $O(k \cdot \log(n))$  to select top-k results from  $n$  documents and thus has the overall time complexity as  $O(k \cdot \log(n) + n)$ . In contrast, MMF only tracks  $Exp\_cum_\tau^k(G_i)$  for top-k results with  $O(|\mathcal{G}| \cdot k)$  time. Besides, it takes  $O(|\mathcal{G}| \cdot k \cdot \log(n))$  time to construct priority queue for each group, and takes  $O(|\mathcal{G}| \times k)$  time to implement Eq.11. The overall complexity of MMF is  $O(k \cdot |\mathcal{G}| \cdot (1 + \log(n)))$ . Therefore, MMF takes  $O((n - |\mathcal{G}|) \cdot k + k \cdot \log(n) \cdot (1 - |\mathcal{G}|))$  less time than FairCo. Because the size of all documents is usually much more than the

**Algorithm 1:** MMF

---

```

1 initialize  $\lambda$  within  $[0,1]$ ,  $k$ ,  $c_t \leftarrow 0$ , initialize  $R^\theta$ ,  $R^{IPS}(d) \leftarrow 0$ ;
2 for each user (time step  $\tau$ ) do
3   if Use_LTR_model then
4     Estimate relevance for all items with model  $R^\theta$ ;
5   else
6     Estimate relevance for all items with  $R^{IPS}(d)$ 
7     according to Eq.5;
8   end
9   for each group  $G_i$  do
10    construct priority queues  $\mathbb{G}_i$  of size  $k$  for group  $G_i$ 
11    with estimated relevance.
12   end
13   ranking=[];
14   for each rank  $i$  do
15     if random  $> \lambda$  then
16        $d_i$  selected with Eq.12, and ranking.append( $d_i$ );
17        $j = G_{ind}(d_i)$ , index group id and assign to  $j$ ;
18        $\mathbb{G}_j.pop()$ 
19     else
20       select  $G_\tau^k$  according to Eq.11 and assign  $G_\tau^k$  to  $j$ ;
21       ranking.append( $\mathbb{G}_j.pop()$ );
22     end
23   end
24   present ranking and collect user clicks  $c_\tau$ , or sampling
25   clicks according to Eq.3;
26   if Use_LTR_model then
27     train  $\tilde{\mathcal{L}}^\tau(\theta)$  loss in Eq.4.;
28   else
29     update  $R^{IPS}(d)$  according to Eq.5;
30   end
31 end

```

---

number of ranks and the number of groups we care (i.e.,  $n \gg k$  and  $n \gg |\mathcal{G}|$ ) in most ranking applications, MMF outperforms FairCo in time.

**4.3.2 Space Complexity.** MMF needs to track of  $Exp\_cum_\tau^k(G_i)$  in Eq. (8) for all positions in top- $k$  rankings for the computation of marginal fairness, which has space complexity as  $O(|\mathcal{G}| \cdot k)$ . FairCo only needs to track the overall cumulative exposure of all documents, which has space complexity as  $O(n)$ . As the number of groups (e.g., race, gender) and the ranks we care are usually small, MMF outperforms FairCo in space.

## 5 EXPERIMENTS AND RESULTS

To evaluate our method, we conduct experiments on one dataset with simulated preference data (i.e., the News dataset [10]) and one dataset with real-world preference data (i.e., the Movie dataset [10]). All the experimental scripts and model implementations used on this paper is available online<sup>2</sup>.

<sup>2</sup><https://github.com/Taosheng-ty/Dynamic-Fairness.git>

### 5.1 Simulated Preference Data

In this paper, we create a simulated preference dataset with the news articles in the AdFrontes Media Bias dataset<sup>3</sup>, which we refer to as the News dataset. In this News dataset, each article contains a polarity value  $\rho^d$  that has been rescaled to the range between -1 and 1 (i.e., left-leaning and right-leaning). Following the methodology used by Morik et al. [20], we simulate a dynamic LTR problem on the News Dataset with simulated users and assign each user with a polarity preference drawn from a mixture of two gaussian distribution and are also clipped to  $[-1, 1]$ .

$$\rho^{u_t} \sim clip_{[-1,1]}(p_{neg}\mathcal{N}(-0.5, 0.2) + (1 - p_{neg})\mathcal{N}(0.5, 0.2)) \quad (15)$$

where  $p_{neg}$  is the probability of the user to be left-leaning. In addition, we simulate each user with an openness parameter  $o^{u_t} \sim \mathcal{U}(0.05, 0.55)$ , indicating on the breadth of interest outside their polarity. With sampled  $u_t$ ,  $\rho^{u_t}$ , and articles' polarity value annotation  $\rho^d$ , we could synthesize a true binary relevance judgement following a Bernoulli distribution as

$$r_t(d) \sim Bernoulli \left[ p = \exp \left( \frac{-(\rho^{u_t} - \rho^d)^2}{2(o^{u_t})^2} \right) \right] \quad (16)$$

In each experiment trial, we sample a set of 30 news articles  $D$  to recommend to users. To investigate ranking fairness, we group articles according to their polarity by assigning articles with  $\rho^d \in [-1, 0]$  to group  $G_1$  and articles with  $\rho^d \in [0, 1]$  to group  $G_2$ .

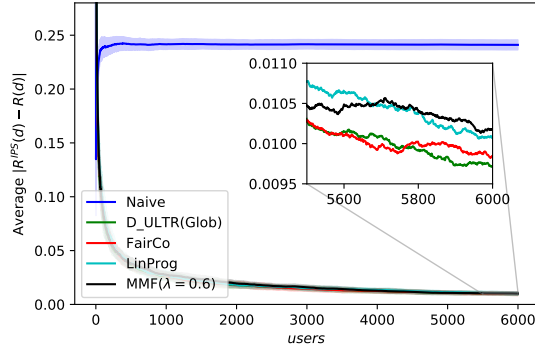
We implement four baselines for comparison. The first one is the **Naive** method that ranks documents by the sum of their observed user clicks (i.e.,  $c_t$ ). The second one is a simple unbiased LTR algorithm **DULTR(Glob)** that ranks documents by the unbiased relevance estimation  $R^{IPS}(d)$  in Eq. (5). To show the effectiveness of MMF as a fairness algorithm, we also include two state-of-the-art fairness algorithms for dynamic LTR, which are **LinProg** [25] and **FairCo** [20]. Proposed by Singh and Joachims [25], LinProg considers group fairness as optimization constraints and implement a linear programming algorithm to maximize ranking fairness based on the relevance estimated by  $R^{IPS}(d)$ . In theory, LinProg would produce the best ranked lists in terms of merit-based fairness. FairCo achieves group fairness by dynamically adding perturbations to the ranking scores of documents based on the exposure of different groups. It has been proven to be effective in optimizing the fairness of dynamic LTR and much more efficient than LinProg.

For all models in the experiments on the News dataset, we use  $R^{IPS}(d)$  to estimate document relevance directly as all users are simulated. This means *Use\_LTR\_model* in Algo.1 is set as False. LinProg and FairCo use a hyper-parameter to reduce unfairness of exposure over all documents. For simplicity, we set it as 0.1 for LinProg and 0.01 for FairCo as these were the tuned hyper-parameters in Morik et al. [20]. For MMF, we tune  $\lambda$  from 0 to 1 and show the corresponding results in Table 2a and Figure 4. Besides, We follow the experiment setting in [20] and adopt the discount function of NDCG as the user's user examination probability for each position for this paper in simulation,

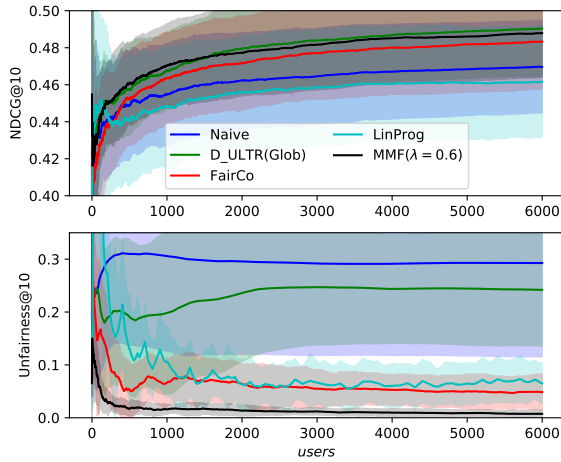
$$p_i = \left( \frac{1}{\log 2(1+i)} \right) \quad (17)$$

where  $p_i$  indicates examination probability at rank  $i$ .

<sup>3</sup><https://www.adfontesmedia.com/interactive-media-bias-chart/>



**Figure 1: The absolute difference between estimated global relevance and true global relevance on News (20 trials).**

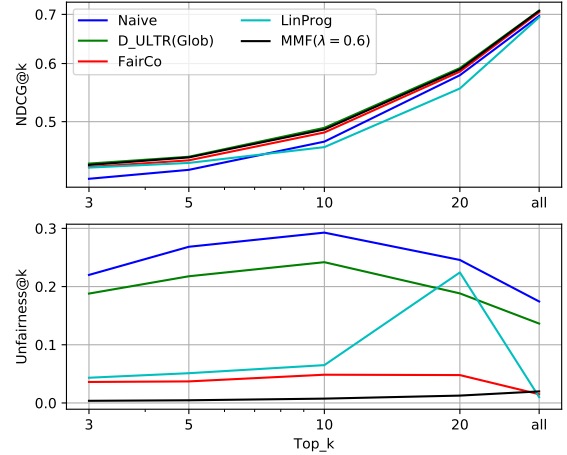


**Figure 2: Convergence of NDCG@10 and Unfairness@10 as the number of users increases on News. (20 trials)**

## 5.2 Results with Simulated Preference Data

**5.2.1 Do the unbiased estimates converge to the true relevance no matter fairness controlling is applied or not?** Fig. 1 shows the absolute difference between estimated global relevance and true global relevance defined in Eq. (5) after applying different ranking algorithm. The error of all algorithm with IPS weighting (Linprog, FairCo, D\_ULTR(Glob) and MMF( $\lambda = 0.6$ )) gradually approach zero, while the error of Naive algorithm keeps around 0.25. There are no significant difference of convergence between IPS weighted algorithms which means we could exclude the influence of relevance estimation when comparing ranking and fairness controlling for IPS weighted algorithms. The result verifies that IPS weighting help to get an unbiased estimation of the true expected relevance for each news articles no matter what ranking algorithm is used and thus estimated relevance can be used to maximize ranking and fairness utility.

**5.2.2 Can MMF effectively reduce unfairness while maintaining good ranking performance?** Fig. 2 shows the convergence NDCG@10 and unfairness@10 for Naive, D-ULTR(Glob), FairCo, Linprog and MMF ( $\lambda = 0.6$ ). Among them, Naive shows the highest unfairness, while NDCG remains low as user interaction increases. D-ULTR(Glob) achieve the best NDCG while its unfairness is worse than all other algorithms except Naive. Our algorithm MMF manages to substantially reduce unfairness while achieve similar NDCG



**Figure 3: Performance of NDCG(top) and Unfairness (bottom) for different prefixes on News (20 trials).**

performance with D\_ULTR(Glob). For the other two algorithm, LinProg and FairCo both show inferior performance than MMF in terms of NDCG and unfairness.

**5.2.3 How MMF performs at different prefixes of a ranking?** Fig. 3 shows the performance NDCG and unfairness at different prefixes for Naive, D-ULTR(Glob), FairCo, Linprog and MMF( $\lambda = 0.6$ ) after 6000 users interactions. More details are shown in Table 2a. In terms of NDCG performance, Naive shows worse performance than IPS weighting algorithms, especially in top ranks, which make sense since richer-get-richer dynamics can be more severe in top ranks. In terms of fairness, among all algorithms, Naive shows the worst performance almost for top ranks. Among the fairness controlling algorithms, Linprog and FairCo show superior performance on overall fairness (i.e., unfairness@all). As is shown in Table 2a, compared to Linprog and FairCo, MMF performs much better on top ranks while showing a slight compromise on overall unfairness. This may attribute to that MMF try to mitigate unfairness for each prefix, while Linprog and FairCo always consider all documents. We think this is the key advantage of our algorithm compared with Linprog and FairCo since top ranks are relatively important for rankings, and many users would leave after viewing top results.

**5.2.4 How does  $\lambda$  control trade-off between ranking performance and unfairness for MMF?** Fig. 4 shows the performance NDCG@10 and unfairness@10 for MMF with different  $\lambda$  after 6000 users interactions. Note that we only show LinProg and FairCo with the tuned hyper-parameters as shown in [20]. Thus their performances are constants. As we can see, the use of hyper-parameter  $\lambda$  in MMF enables us to explicitly control the trade-off between relevance and fairness. When  $\lambda = 0.0$ , MMF degenerates to D\_ULTR(Glob), thus it has similar unfairness with D\_ULTR(Glob), greater than Linprog and FairCo. With the increasing of  $\lambda$ , unfairness gradually decreases and become less than Linprog and FairCo on top results. From the figure, we can see that choosing  $\lambda$  from 0.4 to 0.7 for MMF can achieve better performance than the baseline algorithms on both relevance and fairness.

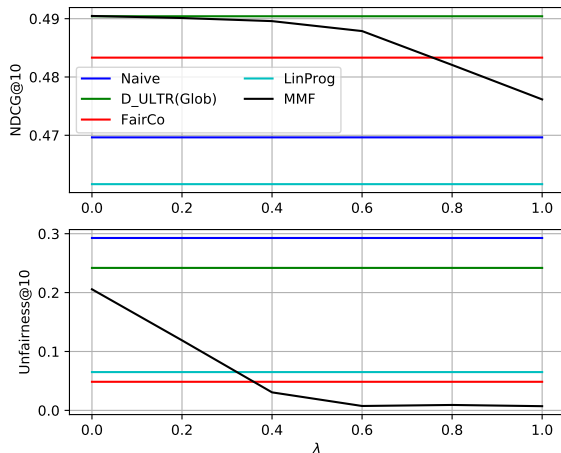
**Table 2: Comparison of MMF with different baselines on News and Movie. Significant improvements or degradations with respect to the performance of FairCo are indicated with +/- in the paired t-test with p-value  $p \leq 0.05$ . The best performance of fair algorithms in each column is highlighted in boldface.**

**(a) Performance of learning-to-rank algorithms on News data.**

		NDCG@3	NDCG@5	NDCG@10	NDCG@all	Unfairness@3	Unf.@5	Unf.@10	Unf.@all
Unfair Algorithms	Naive	0.418 <sup>-</sup>	0.430 <sup>-</sup>	0.470 <sup>-</sup>	0.697 <sup>-</sup>	0.220 <sup>-</sup>	0.268 <sup>-</sup>	0.293 <sup>-</sup>	0.174 <sup>-</sup>
	D_ULTR(Glob)	0.438 <sup>+</sup>	0.448 <sup>+</sup>	0.490 <sup>+</sup>	0.708	0.188 <sup>-</sup>	0.218 <sup>-</sup>	0.242 <sup>-</sup>	0.136 <sup>-</sup>
Fair Algorithms	FairCo	0.434	0.443	0.483	0.705	0.036	0.037	0.049	0.015
	LinProg	0.433	0.439	0.462	0.694	0.043	0.051	0.065	<b>0.010</b>
	MMF( $\lambda = 0.6$ )	<b>0.436</b>	<b>0.447<sup>+</sup></b>	<b>0.488<sup>+</sup></b>	<b>0.708</b>	<b>0.004<sup>+</sup></b>	<b>0.005<sup>+</sup></b>	<b>0.007<sup>+</sup></b>	0.020

**(b) Performance of learning-to-rank algorithms on Movie data.**

		NDCG@3	NDCG@5	NDCG@10	NDCG@all	Unfairness@3	Unf.@5	Unf.@10	Unf.@all
Unfair algorithm	Naive	0.633 <sup>-</sup>	0.638 <sup>-</sup>	0.652 <sup>-</sup>	0.761 <sup>-</sup>	0.166 <sup>-</sup>	0.168	0.111 <sup>+</sup>	0.265 <sup>-</sup>
	D_ULTR(Glob)	0.671 <sup>-</sup>	0.665 <sup>-</sup>	0.672 <sup>-</sup>	0.775 <sup>-</sup>	0.102 <sup>+</sup>	0.105 <sup>+</sup>	0.066 <sup>+</sup>	0.243 <sup>-</sup>
	D_ULTR	0.827 <sup>+</sup>	0.819 <sup>+</sup>	0.816 <sup>+</sup>	0.871 <sup>+</sup>	0.035 <sup>+</sup>	0.036 <sup>+</sup>	0.042 <sup>+</sup>	0.232 <sup>-</sup>
Fair algorithm	FairCo	<b>0.814</b>	0.802	0.791	0.850	0.123	0.162	0.234	<b>0.021</b>
	MMF( $\lambda = 0.1$ )	0.810	<b>0.804</b>	<b>0.802<sup>+</sup></b>	<b>0.863<sup>+</sup></b>	<b>0.008<sup>+</sup></b>	<b>0.011<sup>+</sup></b>	<b>0.016<sup>+</sup></b>	0.312 <sup>-</sup>



**Figure 4: Performance of NDCG@10 (top) and Unfairness@10 (bottom) of MMF with different  $\lambda$  on News (20 trials).**

### 5.3 Real-World Preference Data

To evaluate our method on a real-world preference data, we use the ML20M dataset, which we refer to as the Movie dataset. Following the preprocessing method in [20], we select five production companies with the most movies in the dataset (MGM, Warner Bros, Paramount, 20th Century Fox, Columbia). We aim to ensure fairness of exposure for films from the five production companies, which means movies from the same company belong to the same group. A set of 300 most rated movies by those production companies are selected. Then the 100 movies with the highest standard deviation in the rating across users are selected. For users, we select  $10^4$  users who have rated the most number of the chosen 100 movies. Finally, we get a partially filled rating matrix with  $10^4$  users and 100 movies. We use an off-the-shelf matrix factorization algorithm<sup>4</sup> to fill the missing entries. We then normalize the rating to  $[0, 1]$  by applying a Sigmoid function centered at rating  $b = 3$  with slope  $a = 10$ . Thus, it can serve relevance probabilities where higher star ratings

<sup>4</sup>Surprise library (<http://surpriselib.com/>) for SVD with biased=False and D=50

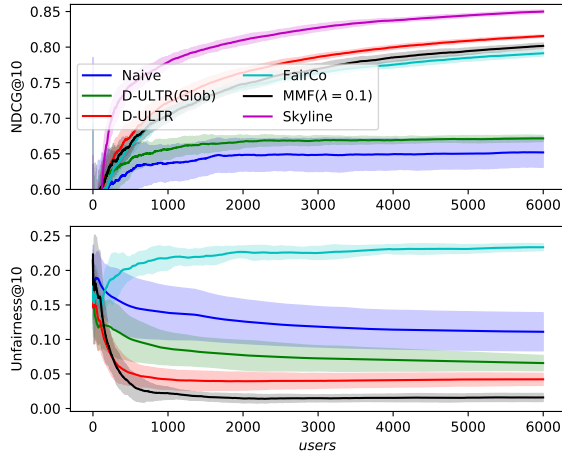
correspond to higher likelihoods of positive feedback, which is used to generate clicks. We use the user embedding from the matrix factorization model as user features  $x_t$ . We keep the dimension of user features to 50. At each time step  $t$ , we sample a user  $x_t$  and the ranking algorithm presents a ranking of the 100 movies. For the user personal relevance estimation model  $R^\theta$  used by FairCo and DULTR, and our method, we use a one hidden-layer neural network that consists of  $D = 50$  input nodes, which corresponds to user feature dimension, then fully connected to 64 nodes in the hidden layer with RELU activation, which is then connected 100 output nodes with Sigmoid to output the predicted relevance probability for the 100 selected movies.

Besides the baselines discussed in Section 5.1, we also include a new baseline in the experiments on the Movie dataset, which is **D-ULTR**. D-ULTR conducts unbiased learning to train the LTR model with click data. Different from D-ULTR(Glob) that directly ranks documents with  $R^{IPS}(d)$ , D-ULTR can model personalized relevance by taking user features into account when building the LTR model. Thus, we expect D-ULTR to perform much better than D-ULTR(Glob) on the Movie dataset with real user preferences. Similarly, we train LTR models with user features for both FairCo and MMF. Note that we exclude LinProg on the Movie dataset as it is designed to work with global relevance and is too computationally expensive for rankings with a large number of documents.

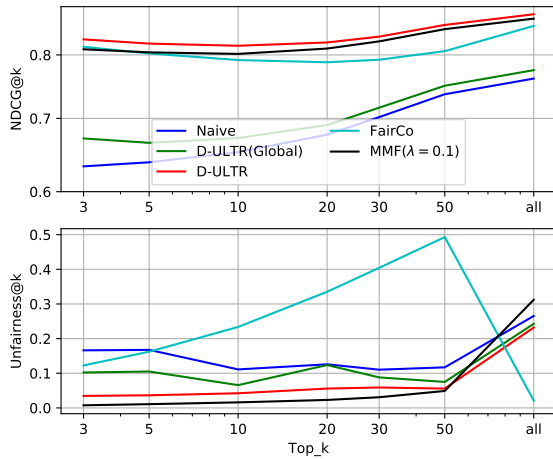
### 5.4 Results with Real-World Preference Data

**5.4.1 Can MMF effectively reduce unfairness while maintaining good ranking performance?** We show the performance of ranking relevance and fairness for all algorithms on the Movie dataset in Fig. 5. For references, we plot a **Skyline** model that trains the LTR model with ground truth relevance judgements and ranks documents via the output estimated relevance from the LTR model. Firstly, as is shown in Fig. 5, personalization do help to reach better ranking performance just as [20] already reported. Ranking algorithms (DULTR, FairCo, MMF, skyline) relying on personalized relevance show superior ranking performance than algorithms like





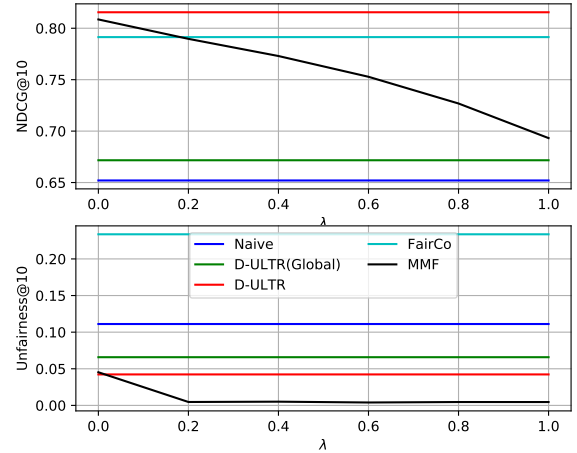
**Figure 5: Convergence of NDCG@10 and Unfairness@10 as the number of users increases on Movie (5 trials).**



**Figure 6: Performance of NDCG(top) and Unfairness (bottom) of MMF with different prefixes on Movie (5 trials).**

Naive and DULTR(Glob), where no personalization is used. Ranking algorithms involving IPS and personalization can approach the skyline as more user interactions are available, which again verifies the effectiveness of IPS and personalization.

After we show the effectiveness of personalization and IPS, we now take a look at the overall performance shown in Table 2b. Naive and DULTR(Glob) show the worst performance in terms of ranking and fairness since no personalization or fairness controlling is involved. Compared to FairCo, our method MMF( $\lambda = 0.1$ ) shows a slight compromise on NDCG@3 while achieving better performance on NDCG@5, NDCG@10, NDCG@all, and much better performance on fairness metrics on top results. The fairness controlling of FairCo is not effective in optimizing fairness on top-k ranks. We think the reason might be that FairCo focuses on mitigating overall unfairness instead of top ranks, which means it ignores fairness of top ranks. Such phenomenon can be clearly shown in Fig. 6.



**Figure 7: Performance of NDCG@10(top) and Unfairness@10(bottom) of MMF with different  $\lambda$  on Movie (5 trials).**

**5.4.2 How does MMF perform at different prefix of a ranking?** Fig.6 shows the performance NDCG and unfairness at different prefixes for Naive, D-ULTR(Glob),ULTR, FairCo,MMF ( $\lambda = 0.1$ ) after 6000 users interactions. As we can see, MMF achieves similar performance with D-ULTR in terms of relevance ranking while maintaining superior performance on ranking fairness on top results. Specifically, it significantly outperforms FairCo on both relevance and fairness for  $k$  from 3 to 50. While FairCo has excellent performance on unfairness@all, it sacrifices both relevance and fairness performance on top results significantly. It even shows more unfairness on top results than unfair algorithms such as D-ULTR.

**5.4.3 How does  $\lambda$  control trade-off of ranking performance and mitigating unfairness for MMF.** Fig. 7 shows the performance NDCG@10 and unfairness@10 for MMF with different  $\lambda$  after 6000 users interactions. Again, we report FairCo with the best parameter settings here. As  $\lambda$  gradually increases, a trade-off can be seen from the growing of fairness (negative unfairness) and decreasing of ranking performance. And we may choose  $\lambda$  to be between 0.0 and 0.2, between which MMF have better ranking performance as well as less unfairness. The optimal  $\lambda$  for Movie data is smaller than for News data. We think it's because of their different relevance distribution. Documents in the News dataset often have similar relevance. In such situation, we should pay more attention to fairness (greater  $\lambda$ ) for News dataset since relevance of different documents are close to each other.

## 6 CONCLUSION AND FUTURE WORK

In this work, we propose a concept of marginal fairness and a Maximal Marginal Fairness (MMF) algorithm for balancing the relevance and fairness of top-k results in dynamic learning to rank. We develop a metric to measure the group fairness of exposure in the top-k results of each ranked list and show that most existing state-of-the-art methods for ranking fairness focus more on the overall fairness of document exposure while compromising a lot in top ranks of each ranked list. In contrast, our proposed MMF algorithm explicitly maximizes the marginal fairness of top-k rankings and can produce better rankings than the state-of-the-art fairness algorithms in both top-k relevance and top-k fairness. In the future,



we will further explore the possibility of extending MMF for more general ranking scenarios or construct new LTR models that integrate the optimization of relevance and fairness from the bottom of model design.

## ACKNOWLEDGMENTS

This work was supported in part by the School of Computing, University of Utah. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

## REFERENCES

- [1] Aman Agarwal, Kenta Takatsu, Ivan Zaitsev, and Thorsten Joachims. 2019. A general framework for counterfactual learning-to-rank. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 5–14.
- [2] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 474–482.
- [3] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 385–394.
- [4] Qingyao Ai, Tao Yang, Huazheng Wang, and Jiaxin Mao. 2020. Unbiased Learning to Rank: Online or Offline? *arXiv preprint arXiv:2004.13574* (2020).
- [5] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *The 41st international acm sigir conference on research & development in information retrieval*. 405–414.
- [6] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.
- [7] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. 2018. Ranking with Fairness Constraints. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- [8] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *Proceedings of the 1st WSDM*. ACM, 87–94.
- [9] Georges E Dupret and Benjamin Piwowarski. 2008. A user browsing model to predict search engine click data from past observations.. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 331–338.
- [10] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.
- [11] Zhengbao Jiang, Ji-Rong Wen, Zhicheng Dou, Wayne Xin Zhao, Jian-Yun Nie, and Ming Yue. 2017. Learning to diversify search results via subtopic attention. In *Proceedings of the 40th international ACM SIGIR Conference on Research and Development in Information Retrieval*. 545–554.
- [12] Jiarui Jin, Yuchen Fang, Weinan Zhang, Kan Ren, Guorui Zhou, Jian Xu, Yong Yu, Jun Wang, Xiaoqiang Zhu, and Kun Gai. 2020. A Deep Recurrent Survival Model for Unbiased Ranking. *arXiv preprint arXiv:2004.14714* (2020).
- [13] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD*. ACM, 133–142.
- [14] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual ACM SIGIR*. Acm, 154–161.
- [15] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2017. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, Vol. 51. Acm New York, NY, USA, 4–11.
- [16] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems* 25, 2 (2007), 7.
- [17] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 781–789.
- [18] Tie-Yan Liu. 2011. *Learning to rank for information retrieval*. Springer Science & Business Media.
- [19] Stepan Malkevich, Ilya Markov, Elena Michailova, and Maarten de Rijke. 2017. Evaluating and Analyzing Click Simulation in Web Search. In *Proceedings of the ACM ICTIR (Amsterdam, The Netherlands) (ICTIR '17)*. ACM, 281–284.
- [20] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (Virtual Event, China) (SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 429–438. <https://doi.org/10.1145/3397271.3401100>
- [21] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for Selection Bias in Learning-to-rank Systems. In *Proceedings of The Web Conference 2020*. 1863–1873.
- [22] Filip Radlinski and Thorsten Joachims. 2006. Minimally invasive randomization for collecting unbiased preferences from clickthrough logs. In *Proceedings of the national conference on artificial intelligence*, Vol. 21. Menlo Park, CA; Cambridge, MA; London; AAI Press; MIT Press; 1999, 1406.
- [23] Stephen E Robertson. 1977. The probability ranking principle in IR. *Journal of documentation* (1977).

- [24] LT Rodrygo, Craig Macdonald, and Iadh Ounis. 2015. Search result diversification. *Foundations and Trends in Information Retrieval* 9, 1 (2015), 1–90.
- [25] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2219–2228.
- [26] Ashudeep Singh and Thorsten Joachims. 2019. Policy learning for fairness in ranking. In *Advances in Neural Information Processing Systems*. 5426–5436.
- [27] Chao Wang, Yiqun Liu, Min Zhang, Shaoping Ma, Meihong Zheng, Jing Qian, and Kuo Zhang. 2013. Incorporating vertical results into search click models. In *Proceedings of the 36th ACM SIGIR*. ACM, 503–512.
- [28] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 115–124.
- [29] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 610–618.
- [30] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 113–122.
- [31] Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, and Xueqi Cheng. 2017. Adapting Markov decision process for search result diversification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 535–544.
- [32] Ke Yang and Julia Stoyanovich. 2017. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. 1–6.
- [33] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th WWW*. ACM, 1011–1018.
- [34] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. Fa\*ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1569–1578.
- [35] Meike Zehlike and Carlos Castillo. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *Proceedings of The Web Conference 2020*. 2849–2855.