UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Fairness of Exposure in Light of Incomplete Exposure Estimation

Heuss, M.; Sarvi, F.; de Rijke, M.

DOI
10.1145/3477495.3531977

Publication date
2022

Document Version
Author accepted manuscript

Published in
SIGIR '22

Link to publication

Citation for published version (APA):
Heuss, M., Sarvi, F., & de Rijke, M. (2022). Fairness of Exposure in Light of Incomplete Exposure Estimation. In *SIGIR '22: proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval : July 11-15, 2022, Madrid, Spain* (pp. 759–769). The Association for Computing Machinery. https://doi.org/10.1145/3477495.3531977

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:10 Oct 2026

Fairness of Exposure in Light of Incomplete Exposure Estimation

Maria Heuss
University of Amsterdam, Amsterdam
The Netherlands
m.c.heuss@uva.nl

Fatemeh Sarvi
AIRLab, University of Amsterdam
Amsterdam, The Netherlands
f.sarvi@uva.nl

Maarten de Rijke
University of Amsterdam, Amsterdam
The Netherlands
m.derijke@uva.nl

ABSTRACT

Fairness of exposure is a commonly used notion of fairness for ranking systems. It is based on the idea that all items or item groups should get exposure proportional to the merit of the item or the collective merit of the items in the group. Often, stochastic ranking policies are used to ensure fairness of exposure. Previous work unrealistically assumes that we can reliably estimate the expected exposure for all items in each ranking produced by the stochastic policy. In this work, we discuss how to approach fairness of exposure in cases where the policy contains rankings of which, due to inter-item dependencies, we cannot reliably estimate the exposure distribution. In such cases, we cannot determine whether the policy can be considered fair. Our contributions in this paper are twofold. First, we define a method called FELIX for finding stochastic policies that avoid showing rankings with unknown exposure distribution to the user without having to compromise user utility or item fairness. Second, we extend the study of fairness of exposure to the top-k setting and also assess FELIX in this setting. We find that FELIX can significantly reduce the number of rankings with unknown exposure distribution without a drop in user utility or fairness compared to existing fair ranking methods, both for full-length and top-k rankings. This is an important first step in developing fair ranking methods for cases where we have incomplete knowledge about the user's behaviour.

CCS CONCEPTS

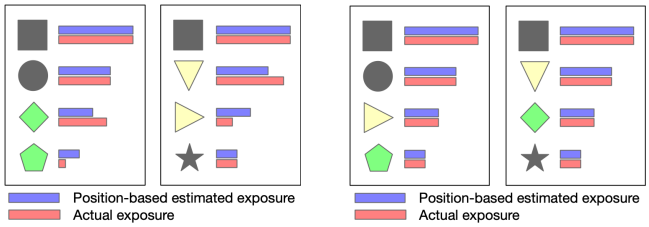
• **Information systems** → *Evaluation of retrieval results*; *Retrieval models and ranking*.

KEYWORDS

Fair ranking; Exposure estimation; Learning to rank

ACM Reference Format:
Maria Heuss, Fatemeh Sarvi, and Maarten de Rijke. 2022. Fairness of Exposure in Light of Incomplete Exposure Estimation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531977

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '22, July 11–15, 2022, Madrid, Spain
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531977



(a) (b)

Figure 1: Visualization of rankings with unknown exposure distribution which are due to inter-item dependencies between items marked by the same color and similar shapes (a). By shuffling some items between rankings in the stochastic ranking policy these dependencies can be reduced such that the estimated exposure agrees with the actual exposure that each item gets (b).

1 INTRODUCTION

There has been increased interest in fair ranking systems, as witnessed by the number of publications [14, 45], the topic's attention during keynotes leading conferences [7, 19], and challenges such as the TREC Fair Ranking track [15]. Several particularities about rankings make this task especially challenging.

First, often ranking systems act as a tool for two-sided marketplaces, such as job markets [17] or music recommender systems [22]. On one side, users want relevant item recommendations. On the other side, items or their providers are interested in being exposed to as many users as possible. Second, biases like position bias can cause a traditional deterministic ranking to amplify small differences in predicted scores into vast differences in user attention [3, 27].

An important line of research on fairness in ranking deals with *fairness of exposure*. Given a ranking, we can estimate how much exposure each item gets in expectation during inference. We call this the *exposure distribution* of the ranking. Singh and Joachims [27] define several notions of fairness of exposure for rankings, among them *disparate treatment*. This notion defines a stochastic ranking policy to be fair if each item or item-group gets expected exposure proportional to its merit. We will mostly focus on individual fairness, where we want to provide each individual item with exposure relative to its merit.

Incomplete exposure estimation. Previous methods for fairness of exposure assume that we can estimate the exposure distribution of any ranking in the set of all possible rankings. For this, a user model like the position-based model [3, 27, 34, 38], or the ERR-based model [12] can be used. However, there are cases where, due to inter-item dependencies that are not accounted for by any of the existing user models, for certain rankings, user-behaviour does not follow the user model; for such rankings we cannot estimate the exposure distribution accurately. See Fig. 1a for an illustration. E.g.,

Sarvi et al. [25] show that visual outliers can have a great impact on the exposure distribution within a ranking, since such outliers attract more user attention. This phenomenon is an example of inter-item dependencies where one item can be perceived as an outlier in the context of items it is presented together with. It can cause the exposure distribution to diverge from the distribution assumed by the user model.

Simply ignoring the incomplete knowledge about the exposure of some of the rankings would imply that we cannot guarantee fairness. Also, by ignoring potentially incomplete exposure estimation, we might introduce a new kind of bias into the collected click data, since items that got more exposure than estimated will have propensity values that are too high, leading to overestimation of their relevance. One solution would be to obtain a more accurate user browsing model by estimating the exposure distribution of rankings that do not follow the user model, through a large-scale user study. To the best of our knowledge no such studies have been conducted. It is also not clear whether one can always reliably estimate the exposure distribution for all possible rankings.

Instead, we propose to avoid showing rankings with unknown exposure distribution to the user by reducing their weight in the probability distribution of the stochastic ranking policy.

Fair top-k ranking. So far, the literature on fairness of exposure has mostly focused on full-length rankings. Top-k rankings are well studied in the general information retrieval (IR) literature [9, 11, 40, 42]; many real-world ranking applications require us to expose just a short list of items. Often there are more relevant items than can be shown to the user, hence it is important to consider fairness of exposure for this set-up as well. Although there have been few approaches to *fair top-k ranking* [42, 44], most are concerned with demographic parity, rather than merit-based fairness of exposure.

Our contributions. In this work we develop a method to find ranking policies that avoid presenting rankings with unknown exposure distribution, while still optimizing for user utility and fairness. Under the assumption that inter-item dependencies are the reason for the shift in exposure, our method works by shuffling items between different rankings to avoid presenting them in a context where they disturb the position-based exposure distribution, as illustrated in Fig. 1b.

We also present what we believe to be the first approach towards fairness of exposure in the top-k setting for the convex optimization approach towards fairness. We generalize the Birkhoff-von Neumann theorem and use this to extend [27] to the top-k setting.

To summarize, our main contributions are as follows:

- We introduce the task of fairness of exposure in light of incomplete exposure estimation and define a novel method FELIX that provides us with a fair ranking policy that avoids rankings with unknown exposure distribution.
- To make FELIX applicable to a broader range of use cases, we extend the constrained optimization approach to fairness of exposure to the top-k case.
- We test and compare FELIX on the outlier use case introduced in [25] and show big improvements over other top-k fair ranking methods in terms of effectiveness in avoiding rankings containing outliers, while staying within the fairness constraints.

2 RELATED WORK

Fairness in ranking. For a detailed overview of fair ranking we refer to [14, 45]. Yang and Stoyanovich [39] seem to have been the first to formalize fairness for rankings in a rank-aware manner, by calculating parity for different top-k cut-offs and summing over these values with a rank-based discount. Zehlike et al. [42, 44] discuss representational fairness for top-k rankings and define a re-ranking algorithm that ensures a share of items from the protected groups in every prefix of the top-k, while Celis et al. [8] formulate the problem as a constrained optimization problem. These papers look for a deterministic ranker, not a stochastic ranking policy, and emphasize on representational fairness and demographic parity.

Singh and Joachims [26] introduce the notion of expected exposure and define fairness of exposure with respect to demographic parity and equal opportunity, where the expected exposure is calculated w.r.t. position bias. Later work [27] defines different types of fairness of exposure w.r.t. disparate impact and disparate treatment, and address the task as a constrained optimization problem. Biega et al. [3] define equity of attention as an alternative notion of fairness for rankings that is also based on exposure; they also address the task as a constrained optimization problem. Wang and Joachims [34] also consider fairness of exposure combined with diversity in rankings. We build on [27] and use the non-uniqueness property of the Birkhoff-von Neumann decomposition that is also used in [34] to produce more diverse rankings. Importantly, we reduce the probability that the user is shown a ranking with unknown exposure distribution rather than providing the user with more diverse rankings as in [34].

Another line of research aims to include fairness in the learning process by including a fairness objective in the objective function [12, 28, 33, 43]. Since inter-item relationships are hard to model within the in-processing set-up, in our work we focus on a post-processing method for avoiding rankings with unknown exposure distribution and leave work on in-processing methods for the future.

Another work that looks into the the topic of uncertainty within fair ranking is [29], which explores fairness of exposure when there is uncertainty about the merit. In contrast to this work, we are considering uncertainty about the exposure of certain rankings.

Exposure estimation in ranking. In counterfactual learning to rank (CLTR) true estimation of exposure plays a central role [20]. Early work on CLTR corrects for position bias using exposure, estimated by a click model [10], as the propensity to inversely weight the importance of clicks [20, 36]. More recent work focuses on estimating examination probabilities [1, 2, 16, 31, 32, 37], which also correlates with exposure, correcting for more types of bias. Recent work on learning fair rankings from implicit feedback [38] simultaneously corrects for position bias and implicit biases in the data. There is no prior work on how to adapt these models for the case where certain rankings do not follow the general user model.

Prior work has shown that exposure might be impacted by other factors than just position and the relevance of other items. Yue et al. [41] observe that visual attractiveness can impact the exposure that items get; Sapiezynski et al. [24] acknowledge that the attention that users give to items in a ranking depends on context; and Wang et al. [35] address the impact of click bait items on exposure distribution. Sarvi et al. [25] show that the existence of visual outliers in rankings

can skew the exposure distribution amongst the items, causing outliers to draw more attention than estimated by the position-based user model that non-outlier rankings seem to follow.

In this work, we focus on similar but more general use cases, where due to inter-item relationships the exposure distribution for some rankings differs from the generally assumed distribution, that can be described through existing user models.

3 BACKGROUND

We introduce preliminaries in fair ranking that form the basis for a new method for ranking under fairness constraints, while avoiding to present rankings with unknown exposure distribution.

3.1 Stochastic ranking policies

Depending on the definition of fairness being used, often a single deterministic ranking cannot achieve fairness [3, 12]. Instead, probabilistic rankers can be used to provide a fair distribution of exposure among items. Given a query q and set of candidate items, $\mathcal{D}_q = \{d_i\}_{i=1,\dots,n}$, to be ranked, we define a *stochastic ranking policy* π_q as a probability distribution over all possible rankings $\mathcal{R}_{\mathcal{D}_q}$. That is, π assigns each ranking $\sigma_j \in \mathcal{R}_{\mathcal{D}_q}$ a probability $\pi_q(\sigma_j)$ that it will be shown to the user.

To evaluate the fairness of a ranking policy we determine the *expected exposure* $\epsilon(d_i \mid \pi_q)$ that each item d_i obtains when enough rankings have been presented to users. To compute this, we need to assume a browsing model that explains the probability of a user visiting an item. Diaz et al. [12] adopt user models corresponding to the ranked-based precision (RBP) and expected-reciprocal rank (ERR), while Singh and Joachims [27] use the position-based user model (PBM). We follow the latter, as it is commonly used in the fairness literature [3, 27, 34, 38]. Assuming that the exposure of an item in a ranking, $\epsilon(d_i \mid \sigma)$, is purely based on its position, the *expected exposure* $\epsilon(d_i \mid \pi_q)$ of document d_i for policy π_q can be calculated as:

$$\begin{aligned}\epsilon(d_i \mid \pi_q) &= \mathbb{E}_{\sigma\sim\pi_q}\epsilon(d_i \mid \sigma)\\ &= \sum_{\sigma\in\mathcal{R}_{D_q}} \pi_q(\sigma)\cdot\epsilon(d_i \mid \sigma)\\ &= \sum_{\sigma\in\mathcal{R}_{D_q}} \pi_q(\sigma)\cdot\frac{1}{\log(1+\text{rank}(d_i \mid \sigma))},\end{aligned} \tag{1}$$

where we assume that the exposure can be calculated based on the rank: $\epsilon(d_i \mid \sigma) = v(rank(d_i \mid \sigma))$ with exposure at rank j given by $v(j) = \frac{1}{\log(1+j)}$.

3.2 Fairness of exposure

The definition of what constitutes a fair ranking may vary between application scenarios and types of biases being addressed [45]. We focus on individual fairness, but our approach can easily be extended for group fairness. Our goal is to make sure that similar items receive a similar amount of exposure that is proportional to their merit. The *merit* $u(d \mid q)$ of an item, $d \in \mathcal{D}$, indicates how much exposure it deserves to get from users with respect to query q. We define the merit of an item as its relevance to the query.

The idea of *fairness of exposure* [27] is to provide each item with exposure ϵ that is proportional to its merit:

$$\frac{\epsilon(d_i \mid \pi_q)}{u(d_i \mid q)} = \frac{\epsilon(d_j \mid \pi_q)}{u(d_j \mid q)} \qquad \forall d_i, d_j \in \mathcal{D}. \tag{2}$$

3.3 Finding a stochastic policy under fairness constraints

To be able to satisfy certain fairness constraints, we need to find a stochastic ranking policy (Section 3.1). Singh and Joachims [27] approach the problem by optimizing for user utility under fairness constraints via linear programming. As our method is based on theirs, we introduce it in more detail. For each query q and item $d \in \mathcal{D}$, let $u(d \mid q)$ be its relevance to the user. We define the *utility* U of a ranking policy π_q as the expected utility to the user, when shown a ranking sampled from π_q:

$$\begin{aligned}U(\pi_q) &= \sum_{d\in\mathcal{D}} \epsilon(d \mid \pi_q)\cdot u(d \mid q)\\ &= \mathbb{E}_{\sigma\sim\pi_q}\sum_{d\in\mathcal{D}} \epsilon(d \mid \sigma)\cdot u(d \mid q).\end{aligned} \tag{3}$$

As we assume a position-based user model, $\epsilon(d \mid \sigma)$ is purely dependent on the position of d in the ranking. Therefore, the expected utility U can be calculated based on the probabilities $P_{i,j} = P(d_i \text{ is placed at rank } j)$:

$$\begin{aligned}U(\pi_q) &= \sum_{d_i\in\mathcal{D}}\sum_{j\in\{1,\dots,n\}} P_{i.j}\cdot v(j)\cdot u(d_i \mid q)\\ &= \mathbf{u}^T\mathbf{P}\mathbf{v},\end{aligned} \tag{4}$$

where $n = |\mathcal{D}|$ is the number of items in the ranking, $\mathbf{u}$ the vector containing the merit of each item, $\mathbf{v}$ the vector containing the position bias at each position, and $\mathbf{P} = \{P_{i,j}\}_{i,j=1,\dots,n}$. Singh and Joachims [27] show that the disparate treatment constraint from Eq. (2) can be formulated as a linear constraint in $\mathbf{P}$, which yields a convex optimization problem of the form:

$$\begin{aligned}\mathbf{P} = \text{argmax}_{\mathbf{P}}\ &\mathbf{u}^T\mathbf{P}\mathbf{v}\\ \text{such that } &\mathbb{1}^T\mathbf{P} = \mathbb{1}\\ &\mathbf{P}\mathbb{1} = \mathbb{1}\\ &0 \le P_{i,j} \le 1\\ &\mathbf{P} \text{ is fair.}\end{aligned} \tag{5}$$

A solution $\mathbf{P}$ to this optimization problem is a doubly stochastic matrix, called the *marginal rank probability* (MRP) *matrix*. The solution $\mathbf{P}$ needs to be transformed into an executable stochastic ranking policy. The Birkhoff-von Neumann theorem [5] gives us a constructive proof that such a matrix can be decomposed into a convex sum of $M \le n^2 - n + 1$ permutation matrices:

$$\mathbf{P} = \sum_{m=1,\dots,M} \alpha_m P_{\sigma_m} \text{ such that } \sum_{m=1,\dots,M} \alpha_m = 1\ (0 \le \alpha_m \le 1). \tag{6}$$

Since each permutation matrix corresponds to some ranking, we denote the permutation matrix corresponding to σ by P_σ.

With this we have found a stochastic policy π with $\pi(\sigma_m) = \alpha_m$ and $\pi(\sigma) = 0$ for all σ not contained in this convex sum. Note that this decomposition is not necessarily unique; in Section 4.3 below we will make use of this fact.

3.4 The impact of outliers on the exposure in rankings

Sarvi et al. [25] provide evidence that commonly made assumptions on the user-behaviour might not hold when the presented ranking contains visible outliers that might attract the attention of the user. Since outliers are an example where inter-item dependencies between documents can change the exposure distribution among the items in a ranked list, we work with this example for our experiments in Section 5. We follow the set-up of [25], where the authors assume that outliers can be determined through outlier detection on a specific visual item feature $g(d)$ that might impact the user's perception of an item. In the case of scholarly search, which is used as an example in the experiments, such a feature could be the number of citations that each document has.

Outliers are considered in a context $C \subset \mathcal{D}$ of items that are presented together, which could for instance be the top-k that is presented in a single search engine result page (SERP). Given such a context $C = \{d_1, \ldots, d_k\} \subset \mathcal{D}$, we use the features, $g(d_1), \ldots, g(d_k)$, as input for the outlier detection. Sarvi et al. [25] find that the performance of their method for removing outliers from the rankings is not very sensitive to the outlier detection method. For simplicity, we will therefore use the Z-score:

$$z(g_i) = \frac{g_i - \mu}{s}, \tag{7}$$

where $g_i = g(d_i)$, and $\mu = \frac{1}{k}\sum_{i=1}^{k} g_i$ and $s = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(g_i - \mu)^2}$ denote the mean and standard deviation of the scores in that context. Given these Z-scores, we define an item d_i to be an outlier if $|z(g_i)| > \lambda$, where λ can be chosen dependent on the sensitivity towards outlier items. Here, we diverge slightly from [25], who use a more complex outlier detection method.

Next, we introduce an extension to the convex optimization approach to fairness of exposure from Section 3.3 for top-k rankings. We use the definition of fairness of exposure with respect to disparate treatment from Section 3.2 and work with stochastic policies from Section 3.1. We also develop a method that avoids displaying rankings with unknown exposure distribution, using the outlier use case from Section 3.4 for our experiments in Section 5.

4 FAIRNESS OF EXPOSURE UNDER INCOMPLETE EXPOSURE ESTIMATION

As discussed in Section 3.1, previous work on fair ranking assumes that we can estimate the exposure distribution for all rankings in a policy with one user model. Often, the position-based user model is used. But there are cases where these assumptions do not hold up. Sarvi et al. [25] show that the existence of outliers in a displayed ranking can strongly impact the exposure distribution of the ranking. To the best of our knowledge, there is no prior work on estimating the exposure distribution of such rankings. If such *rankings with unknown exposure distribution* are part of a stochastic ranking policy (i.e., if such a ranking has a non-zero probability of being presented to the user), we cannot determine whether the policy is fair. Therefore, for attaining fair stochastic policies we should avoid using such rankings. This introduces the task of fair ranking under incomplete exposure estimation.

In this section we develop a method for the task of **F**airness of **E**xposure in **L**ight of **I**ncomplete e**X**posure estimation, FELIX, that provides a ranking policy that avoids rankings with unknown exposure distribution without damaging fairness or utility. FELIX is based on the assumption that the shift in the exposure distribution is caused by inter-item relationships between the items that are ranked together. Hence, depending on the context an item is presented in, it could either follow the position-based exposure distribution or it could draw more or less exposure than assumed. In the example, an outlier in a ranking might draw more attention than a non-outlier item at the same position, as demonstrated in [25]. When presented in a more diverse ranking, the same item might not be considered an outlier any more and follow the assumed position-based exposure distribution. Compared to the method for removing outliers from the top-k in [25], FELIX is more generally applicable to any use case where, due to inter-item dependencies, some rankings have unknown exposure distribution. Also, FELIX allows us to consider outliers in the local context that they are presented in, while Sarvi et al.'s approach can only remove outliers with respect to the global context of all items in the list.

Since the context in which items are presented in plays a central role for our task, naturally we are interested in our method to work in the top-k setting. Therefore, we first generalize the constrained optimization approach towards fairness of exposure, introduced in [27], to the top-k setting and present an efficient way to determine a fair policy. Then we present our method FELIX that uses iterative re-sampling to determine a stochastic policy that avoids presenting rankings with unknown exposure distribution to the user, while staying within the fairness constraints.

4.1 Fair ranking in the top-k setting

We will now extend the convex optimization approach to fairness to the top-k setting. Let n be the number of candidate items to be ranked and $k \leq n$ be the number of ranks of the desired rankings. As explained in Section 3.3, searching for a stochastic policy under fairness constraints can be done by first searching for a marginal rank probability matrix $\mathbf{P}$ that satisfies the fairness constraints, and then decomposing this matrix. Since we are interested in the top-k case, $\mathbf{P} = \{P_{i,j}\}_{i=1,\ldots n, j=1,\ldots k}$ is now a $n \times k$ matrix, where $P_{i,j}$ is the probability that item i is placed at rank j. With $\mathbf{u}$ the n-dimensional utility vector and $\mathbf{v}$ the k-dimensional vector containing the examination probability at each of the top-k positions we can solve the following linear program:

$$\begin{aligned} \mathbf{P} = \text{argmax}_{\mathbf{P}}\ & \mathbf{u}^T\mathbf{P}\mathbf{v} \\ \text{such that } & \mathbb{1}_n^T\mathbf{P} = \mathbb{1}_k \\ & \mathbf{P}\mathbb{1}_k \leq \mathbb{1}_n \text{ (element-wise inequality)} \\ & 0 \leq P_{i,j} \leq 1 \\ & \mathbf{P} \text{ is fair.} \end{aligned} \tag{8}$$

Given the marginal rank probability matrix $\mathbf{P}$, we want to determine a stochastic policy given by a distribution over actual rankings. In the $n \times n$ setting, the Birkhoff-von Neumann (BvN) decomposition provides us with an algorithm to determine such a distribution. The following result generalizes the BvN theorem to the $n \times k$ setting where n is not necessarily equal to k.

Theorem 4.1. *Any matrix* $P = \{a_{i,j}\}_{i \le n, j \le k}$ *with* $\forall i, j : 0 \le a_{i,j} \le 1, \forall j : \sum_{i=1}^{n} a_{i,j} = 1$ *and* $\forall i : \sum_{j=1}^{k} a_{i,j} \le 1$ *can be written as the convex sum* $P = \sum_{l=1}^{m} \alpha_l \cdot P_l$ *of permutation matrices* P_l *with coefficients* $\alpha_l \in [0, 1]$ *such that* $\sum_{l=1}^{m} \alpha_l = 1$.

Proof. In Lemma 4.2 below, we show that P can be extended to a doubly stochastic matrix P'. We can use the BvN decomposition for doubly stochastic matrices to find a decomposition for P', which will induce a decomposition for P. For details, see the Appendix. □

Here we say that $P' \in \mathbb{R}^{n' \times k'}$ is an *extension* of $P \in \mathbb{R}^{n \times k}$ if $n' \ge n, k' \ge k$, and $P_{i,j} = P'_{i,j}$ for all (i, j) with $i \le n$ and $j \le k$. We will denote this by $P'|_{i \le n, j \le k} = P$.

Lemma 4.2. *Let* $P = \{a_{i,j}\}_{i \le n, j \le k}$ *be a matrix with the same properties as described in Theorem 4.1 with* $k \le n$. *Then there is a matrix* $P' = \{a'_{i,j}\}_{i \le n, j \le n}$ *with* $\forall i, j : 0 \le a'_{i,j} \le 1$ *such that* $P = P'|_{i \le n, j \le k}$, *and* $\forall i : \sum_{j=1}^{n} a'_{i,j} = 1$ *and* $\forall j : \sum_{i=1}^{n} a'_{i,j} = 1$.

Proof. Define $P' = \{a'_{i,j}\}_{i \le n, j \le n}$ as

$$a'_{i,j} = \begin{cases} a_{i,j} & \text{if } j \le k \\ \frac{1 - \sum_{j'=1}^{k} a_{i,j'}}{n-k} & \text{if } j > k. \end{cases} \tag{9}$$

Then $P'|_{i \le n, j \le k} = P$ by definition. P' satisfies all the requirements from the lemma. A proof of this can be found in the Appendix. □

By transposing A we can show that the Lemma also holds if $k > n$.

4.2 An efficient implementation of the generalized Birkhoff-von Neumann decomposition

For an implementation of the generalized Birkhoff-von Neumann theorem, one can in theory use the proof of Theorem 4.1 and extend the MRP-matrix, that we obtained by solving the convex optimization problem from Eq. 8, to a full $n \times n$-matrix. This matrix can then be decomposed into the convex sum of permutation matrices with help of the BvN theorem for doubly stochastic matrices after which we can restrict the matrices again to the first k columns. Since the complexity of the BvN decomposition for square matrices is $O(n^4\sqrt{n})$ [18, 21] and hence infeasible for large n, we propose an alternative implementation for $n \times k$ or $k \times n$ matrices with $k < n$, that can be implemented with time complexity $O(k^3 n^2)$.

Algorithm 1 gives a structured overview of our algorithm for the generalized BvN decomposition. We start off by noting that the way in which we extended the doubly stochastic matrix from P in the proof of Lemma 4.2 is not unique. For any index pair $(i, j), (i', j')$ with $j, j' > k$ we can subtract some value β from $a'_{i,j}$ and $a'_{i',j'}$, while adding the same value to $a'_{i',j}$ and $a'_{i,j'}$. The resulting matrix will have the same properties as P' and will also be an extension of P. Therefore, instead of extending P to a full doubly stochastic matrix, we can extend it to an $n \times (k+1)$ matrix $\widetilde{P}$, where the last column contains the entries that make the values of each row sum to 1. In the decomposition we split off matrices that are permutation matrices on the first k columns and have $n - k$ non-zero entries on the last column; see line 2 in Algorithm 1.

Algorithm 1 Algorithm for the generalized Birkhoff-von Neumann decomposition.

Require: $P \in \mathrm{Mat}_{n \times k}$ with properties as in Theorem 4.1
1: Initialize $\mathcal{P} = \{\}$ empty decomposition
2: Extend P to $\widetilde{P}$ by adding a column $\{c_i\}_{i=1,\dots,n}$ with values $c_i = 1 - \sum_{j=1}^{k} P_{i,j}$
3: **while** $\widetilde{P} \neq 0$ **do**
4: Translate $\widetilde{P}$ to a bipartite graph with n resp. $k+1$ vertices on each side with edges between the i-th and j-th vertex if $P_{i,j} \neq 0$
5: Find a perfect matching m (with multiplicity of $n - k$ for the last vertex) with the adjusted Hopcroft-Karp algorithm
6: Translate m to a matrix P^m, where $P^m|_{i \le n, j \le k}$ forms a permutation matrix.
7: $\alpha = \min_{\{i,j | P^m_{i,j} \neq 0\}} (\widetilde{P}_{i,j})$
8: $\mathcal{P} \leftarrow \mathcal{P} + (\alpha, P^m|_{i \le n, j \le k})$
9: $\widetilde{P} \leftarrow \widetilde{P} - \alpha P^m$
10: **end while**
11: Return $\mathcal{P}$

We can use this realization to extend the implementation of the BvN algorithm [4], which translates the marginal rank probability matrix into a bipartite graph and uses the Hopcroft-Karp algorithm [18] to find a perfect matching m, which in turn can be translated back into a permutation matrix, P^m; see line 4, 5 and 6.[1]

In the next step, line 7, we calculate the biggest coefficient α, such that subtracting the scaled permutation matrix αP^m, still results in a matrix with only non-negative coefficients. We add the coefficient-matrix pair to the decomposition and subtract the scaled permutation matrix from $\widetilde{P}$; see line 8 and 9. By translating the matrix $\widetilde{P}$ into a bipartite graph, where the node corresponding to the $(k+1)$-th column has multiplicity $n-k$, and adjusting the Hopcroft-Karp algorithm (line 5) slightly to allow for certain vertices to be matched with higher multiplicity, we can significantly speed up this part of the algorithm from $n^2\sqrt{n}$ to $k^2 n$. Since the upper bound of matrices in the decomposition decreases from order n^2 to kn the complexity changes as stated in the following Theorem. A proof of this statement can be found in the Appendix A.2

Theorem 4.3. *Using the modified top-k algorithm for the generalized Birkhoff-von Neumann theorem, Algorithm 1, a decomposition as described in Theorem 4.1 can be obtained with time complexity* $O(k^3 n^2)$.

4.3 Determining a stochastic policy that avoids rankings with unknown exposure distribution

As explained in Section 3.4, certain types of rankings can have a non-typical exposure distribution. Allowing such rankings invalidates the approach by Singh and Joachims [27], since a position-based exposure vector $\mathbf{v}$ is used in both the utility calculation and the fairness constraint in their approach. In this section our goal is to find a stochastic policy that avoids rankings for which the exposure distribution is unknown. We will use a re-sampling strategy,

[1] For the implementation we used https://networkx.org and https://github.com/jfinkels/birkhoff

which, after the decomposition step in Eq. 6, rejects rankings with unknown exposure distribution. The core idea we present below is based on the assumption that the inter-item dependencies between some of the items is the cause of the shift in exposure and that by shuffling the items between different rankings, rankings with unknown exposure distribution might be changed into rankings with known exposure distribution.

Algorithm 2 gives a step-by-step overview of the algorithm used by FELIX. Similarly to Wang and Joachims [34], we make use of the fact that the Birkhoff-von Neumann decomposition is not unique. For most doubly stochastic matrices there is a large number of possible decompositions [13], which makes it possible for us to search for a decomposition that does not have a lot of weight on rankings with unknown exposure distribution. After determining the MRP matrix $\mathbf{P}$ (line 1), we decompose it into the sum $\mathbf{P} = \sum_{i=1}^{M} \alpha_i P_{\sigma_i}$. In the top-$k$ setting this can be done by using the generalized Birkhoff-von Neumann algorithm (Algorithm 1); see Algorithm 2 line 4. We write $\mathcal{P} = \{(\alpha_i, P_{\sigma_i})\}_{i=1,\dots,M}$ for the set of coefficient, matrix pairs in this convex sum. Once the matrix is fully decomposed, we divide the resulting coefficient, permutation matrix pairs (α_i, P_{σ_i}) into two groups, one containing all the permutations where the corresponding ranking has a known exposure distribution amongst its items and the other one containing pairs corresponding to rankings with unknown exposure distribution:

$$\mathcal{P}_{known} = \{(\alpha_i, P_{\sigma_i}) \in \mathcal{P} | \sigma_i \text{ has known exposure distribution}\}$$
$$\mathcal{P}_{unknown} = \mathcal{P} - \mathcal{P}_{known}.$$

We use the elements of $\mathcal{P}_{known}$ directly as a part of the final decomposition; see lines 5–7. The elements of $\mathcal{P}_{unknown}$ are aggregated, weighted by their coefficient; see line 8.

$$\widetilde{\mathbf{P}} = \sum_{(\alpha_i, P_i) \in \mathcal{P}_{unknown}} \alpha_i \cdot P_i. \tag{10}$$

Up to scalar multiplication, the resulting matrix $\widetilde{\mathbf{P}}$ satisfies the required characteristics of Theorem 4.1 and hence can be decomposed again with the generalized BvN decomposition (Algorithm 1).

This decomposition-aggregation process repeats for a number of iterations, *iter* (line 3–10). In each iteration, the recombination of rankings with unknown exposure distribution makes it possible for the algorithm to group items together that previously have not been together in one ranking. Through this re-sampling, the context in which items are presented changes, which often also means that the exposure distribution of these newly ranked list is known. Note that this approach does not remove items from the rankings, but rather shuffles the items among different rankings within the decomposition. After *iter* iterations the remaining rankings with unknown exposure distribution are being added to the policy (line 11–13) to ensure the fairness and utility, that was optimized for.

4.4 Upshot

To summarize Section 4, we extended the continuous optimization approach to fairness for the top-k setting in Section 4.1 by proving that the Birkhoff-von Neumann theorem, which is used to decompose the matrix that was attained through the convex optimization, can be extended to a more general setting. In Section 4.2 we gave an algorithm for the decomposition in the top-k case and discussed an efficient implementation. This extends the space of use cases to which this approach to fair ranking can be applied. We will use this in our experiments, which will partly be conducted in the top-k setting. In Section 4.3 FELIX is introduced, which, by iteratively rejecting rankings with unknown exposure distribution, reduces the probability that such rankings are shown to the user.

Next, we test the performance of the proposed method for top-k fairness. Furthermore, we investigate how well FELIX is able to avoid rankings with unknown exposure distribution and how this impacts the performance w.r.t. fairness and user utility.

Algorithm 2 Fairness of Exposure in Light of Incomplete Exposure Estimation (FELIX)

Require: $\mathcal{D}_q$, k, merit vector $\mathbf{u}$, position bias vector $\mathbf{v}$, number of iterations *iter*

1: Determine MRP matrix $\mathbf{P}$ as in Eq. 8 with $\mathbf{u}$ and $\mathbf{v}$
2: Initialize $\pi(\sigma) = 0, \forall \sigma \in \mathcal{R}_{\mathcal{D}}$
3: **while** *iter* $\neq 0$ **do**
4: $\quad \mathcal{P} \leftarrow$ Decompose $\mathbf{P}$ with Algorithm 1
5: $\quad$ **for all** $(\alpha, P_\sigma) \in \mathcal{P}_{known}$ **do**
6: $\quad\quad \pi(\sigma) \leftarrow \pi(\sigma) + \alpha$
7: $\quad$ **end for**
8: $\quad \mathbf{P} \leftarrow \sum_{(\alpha, P_\sigma) \in \mathcal{P}_{unknown}} \alpha \cdot P_\sigma$
9: $\quad$ *iter* $\leftarrow$ *iter* $- 1$
10: **end while**
11: **for all** $(\alpha, P_\sigma) \in \mathcal{P}_{unknown}$ **do**
12: $\quad \pi(\sigma) \leftarrow \pi(\sigma) + \alpha$
13: **end for**
14: Return π

5 EXPERIMENTAL SET-UP

We experiment with two variants of our model: to evaluate our top-k approach to fair ranking we use FELIX without re-sampling i.e., with only one iteration, denoted by $\textbf{FELIX}_{iter=1}$; to evaluate our method for reducing the probability of generating rankings with unknown exposure we use 20 iterations ($\textbf{FELIX}_{iter=20}$).

Our experiments aim to answer the following research questions: (RQ1) Can $\textbf{FELIX}_{iter=1}$ provide fair top-k rankings while maintaining the user utility compared to the baselines? (RQ2) Can $\textbf{FELIX}_{iter=20}$ reduce the probability of showing rankings with unknown exposure distribution to the user without compromising fairness or utility, compared to other methods? We use the case of rankings with outliers as an example for rankings with unknown exposure distribution. As Sarvi et al. [25] show, outliers can change the exposure distribution that items collect in expectation; we broadly follow their experimental set-up to be able to compare to prior work that is, for this specific use case, closest to our approach.

Datasets. Our experiments in Section 6 use two academic search datasets provided by the TREC19 and TREC20 Fair Ranking track.[2] These datasets come with queries, relevance judgements, and information about the authors and academic articles extracted from the Semantic Scholar Open Corpus.[3] See Table 1 for descriptive statistics of the datasets. Since we experiment on the task of removing outliers from the top-k, which only makes sense for queries with

[2] https://fair-trec.github.io/
[3] http://api.semanticscholar.org/corpus/

Table 1: Descriptive statistics of the original and pre-processed TREC Fair Ranking track 2019 and 2020 data.

	2019		2020	
	Train	Test	Train	Test
Avg. list size (original)	4.1	4.1	23.5	23.4
Avg. list size (pre-proc.)	4.1	13.0	23.5	31.9
Avg. # rel. items/list (original)	2.0	2.0	3.7	3.4
Avg. # rel. items/list (pre-proc.)	2.0	4.4	3.7	4.5

enough items, for testing we only use rankings with at least 20 items. The 2020 dataset comes with 200 queries for training and 200 for testing; keeping only the lists with at least 20 papers leaves us with 112 test queries. Similarly, the 2019 dataset comes with 631 queries for training and 631 for testing. However the test set contains only 3 queries with more than 20 items, which is not acceptable. As a pragmatic solution, we keep lists with at least 10 items, which leaves us with 69 test queries, but up-sample each of these queries to 50 items by using the feature vectors of non-relevant items from other random lists as negative samples.

Experiments. We consider approaches where correcting for fairness is a post-processing step. We use ListNet [6] as our learning to rank (LTR) model for the ranking step, with a maximum of 30 epochs, the Adam optimizer with learning rate of 0.02, and early stopping. As input to the LTR model we use the same data as OMIT[4] with 25 features based on term frequencies, BM25 [23], and language models [30, 46].[5]

To be able to treat the output of the LTR model as the relevance probabilities we normalize the predicted scores to be within the range $[\epsilon, 1]$ with $\epsilon = 10^{-4}$. Choosing $\epsilon > 0$ ensures that each item has a non-zero probability of being placed in a ranking.

As mentioned earlier in this section, we use rankings that contain visible outliers as example for rankings with unknown exposure distribution. Following [25] we use the number of citations of a paper as a visible feature that may be subject to outliers. For the context in which outliers are perceived we use the top-k items. We use the Z-score with threshold value 2.5 to determine whether an item can be considered an outlier; see Section 3.4.

We conduct two types of experiments. The first experiment imitates the experimental set-up of Sarvi et al. [25], where full rankings are formed but the presence of outliers is only measured in the top-k of each ranking. The second experiment looks at top-k ranking. We use $k = 10$ in our experiments and aim for individual fairness as opposed to [25, 27], where group fairness is used.

Baselines. To answer research questions (RQ1) and (RQ2), we compare **FELIX**$_{iter=1}$ and **FELIX**$_{iter=20}$ with the following baselines:

PL As suggested in [12], we use a Plackett-Luce (PL) ranker initialized with the predicted, normalized scores of the LTR model.

PL-random We use a PL ranker over a uniform score distribution as a baseline for a random ranker.

Vanilla We use the method introduced by Singh and Joachims [27] with only fairness constraints as the vanilla baseline. This is the model we build upon.

Deterministic This baseline is ListNet, our traditional LTR model.

OMIT The method introduced in [25], where a similar optimization problem is solved as for Vanilla, but with an additional regularizing objective that punishes rankings with a global outlier in the top-k.

For the experiments on the top-k, we only sample $k = 10$ items from the PL models, **PL@10** and **PL-random@10**. Since **FELIX**$_{iter=1}$ is a novel extension of the Vanilla convex optimization approach for the top-k setting, we do not have the Vanilla baseline in this setting. For OMIT we use our top-k convex optimization approach with the additional outlier objective, **OMIT@10**, to be able to compare the outlier reduction of FELIX$_{iter=20}$ and OMIT in the top-k setting.

Evaluation. To evaluate fairness we use the EE-L metric [12]. The target exposure of item d_i is calculated as $\epsilon^*(d_i) = \epsilon_{total} \cdot u(d_i)/\sum_j u(d_j)$, where ϵ_{total} is the total amount of exposure that users spend in expectation on the ranking, and $u(d_i)$ is the merit, i.e. relevance, of item d_i. Given the expected exposure of all items as a vector ϵ, the *expected exposure loss*, *EE-L* can be calculated as:

$$\text{EE-L} = \ell\left(\epsilon, \epsilon^*\right) = \left\|\epsilon - \epsilon^*\right\|_2^2. \tag{11}$$

Ranking utility performance is measured with NDCG.

For a given query, to evaluate how well a policy π performs in avoiding rankings with unknown exposure distribution, we measure the probability that such a ranking is displayed by the policy. In our experiments this translates to measuring the probability that a randomly sampled ranking, σ contains an outlier:

$$\begin{aligned} P(u \mid \pi) &= P(\sigma \text{ has unknown exposure distribution} \mid \sigma \sim \pi) \\ &=_{\text{here}} P(\#\text{ outliers in } \sigma \geq 1 \mid \sigma \sim \pi). \end{aligned}$$

Additionally, for comparability with [25], we measure:

$$\text{Outlierness}@k(\pi) = \mathbb{E}_{\sigma\sim\pi} \sum_{d_i \in \text{top-}k(\sigma)} \mathbb{1}(d_i \text{ is outlier}) z(d_i).$$

For each metric we report the average value taken over all queries. Each experiment was conducted 5 times with different train/validation split and different random seed. Each split uses 80% of the train-data for training and 20% of the train-data for validation. In our result tables we report the mean results. We test for significance with a two tailed paired students t-test, using the metric values over all queries as input and comparing each method with FELIX$_{iter=20}$.

6 RESULTS

Table 2 and 3 contain the results for our experiments on the top-k and full ranking set-up, respectively.

(RQ1): Can FELIX$_{iter=1}$ provide fair top-k rankings while maintaining the user utility compared to the baselines? To answer this research question we first compare the performance of FELIX$_{iter=1}$ with PL@10, since this is the only baseline that has as its objective to create fair top-k ranking policies. For both utility and fairness FELIX$_{iter=1}$ performs marginally better on TREC20 data. In the case of TREC19 data, FELIX$_{iter=1}$ still has slightly better user utility; the fairness scores are close to identical. Overall none of these differences are significant.

As a sanity check, looking at our other baselines, we see that w.r.t. user utility (NDCG), in Table 2 the deterministic ranker outperforms all probabilistic rankers, which is expected since it is purely optimized for utility. This is reflected in the fairness score, where the

[4] https://github.com/arezooSarvi/OMIT_Fair_ranking
[5] Our experimental code is based on https://github.com/MilkaLichtblau/BA_Laura.

Table 2: Top-k rankings. Significance is measured with a two-tailed paired t-test; all comparisons are against FELIX$_{iter=20}$.

	Method	Optimizing Fairness	NDCG↑ @5	NDCG↑ @10	Fairness↓ EE-L	$P(u \mid \pi)$↓ @10	Outlierness↓ @10
TREC20	**FELIX$_{iter=20}$**	Yes	0.203	0.279	**6.22**	**0.20**	**0.115**
	FELIX$_{iter=1}$	Yes	0.203	0.279	6.23	0.39*	0.151*
	PL@10	Yes	0.197	0.275	6.24	0.47*	0.174*
	PL-random@10	No	0.177*	0.249*	6.29	0.47*	0.175*
	Deterministic	No	**0.287***	**0.370***	7.22*	0.41*	0.154*
	OMIT@10	Yes	0.198	0.273	6.34	0.33*	0.132*
TREC19	**FELIX$_{iter=20}$**	Yes	0.12	0.16	5.9	**0.12**	**0.08**
	FELIX$_{iter=1}$	Yes	0.12	0.16	5.9	0.30*	0.12*
	PL@10	Yes	0.11	0.16	**5.8**	0.35*	0.14*
	PL-random@10	No	0.10	0.15	**5.8**	0.41*	0.16*
	Deterministic	No	**0.15**	**0.21***	7.5*	0.25*	0.12*
	OMIT@10	Yes	0.11	0.15	6.0	0.23*	0.10

Table 3: Full length rankings, remove outliers from the top-k. Significance is reported in the same way as in Table 2.

	Method	Optimizing Fairness	NDCG↑ @5	NDCG↑ @10	Fairness↓ EE-L	$P(u \mid \pi)$↓ @10	Outlierness↓ @10
TREC20	**FELIX$_{iter=20}$**	Yes	0.221	0.302	**24.5**	**0.24**	**0.126**
	Vanilla	Yes	0.221	0.302	**24.5**	0.40*	0.163*
	PL	Yes	0.192*	0.269*	24.7	0.45*	0.169*
	PL-random	No	0.178*	0.249*	24.9	0.47*	0.175*
	Deterministic	No	**0.267**	**0.348**	24.7	0.40*	0.152
	OMIT	Yes	0.221	0.302	**24.5**	0.34*	0.139
TREC19	**FELIX$_{iter=20}$**	Yes	0.15	0.22	**46.4**	**0.11**	**0.06**
	Vanilla	Yes	0.16	0.22	**46.4**	0.14	0.07
	PL	Yes	0.12	0.17	**46.4**	0.32*	0.13*
	PL-random	No	0.10*	0.15*	46.5	0.41*	0.16*
	Deterministic	No	**0.17**	**0.23**	46.6	0.12	0.07
	OMIT	Yes	0.13	0.18	46.5	0.15	**0.06**

deterministic ranker scores significantly worse than FELIX$_{iter=20}$. W.r.t. utility, the random ranker is outperformed by all other probabilistic ranking methods, showing that these methods present users with better results than a uniform ranking policy would.

To summarize, we find no significant differences in terms of utility or fairness between FELIX$_{iter=1}$ on the one hand and the PL-ranker on the one hand. This makes our approach suitable for top-k ranking under fairness constraints and hence allows us to extend FELIX for this setting. In the rest of this section, we will see other advantages of FELIX over the PL baseline.

(RQ2): Can FELIX$_{iter=20}$ reduce the probability of showing rankings with unknown exposure distribution to the user, without having to compromise fairness or utility, compared to other methods? We are interested in the trade-offs between user utility, fairness and the probability of showing rankings with unknown exposure, which is indicated by $P(u \mid \pi)$, in Tables 2 and 3. For the TREC20 data, in both settings FELIX$_{iter=20}$ successfully improves $P(u \mid \pi)$ while maintaining the NDCG@10 and EE-L scores compared to all baselines. Our main baseline to compare with for this research question is OMIT, as it is the only model that optimizes for presenting fewer outliers in the top-k positions. Compared to OMIT, FELIX$_{iter=20}$ achieves significantly better results in terms of $P(u \mid \pi)$ for both settings, while keeping the same (or better) scores for other metrics. For the top-k experiment, we also see a significant improvement w.r.t. $P(u \mid \pi)$, compared to FELIX$_{iter=1}$: iteratively re-sampling successfully reduces the number of rankings with unknown exposure distribution in the policy. For the TREC19 data we can still observe that FELIX$_{iter=20}$ offers the best trade-off between the three objectives in the top-k setting. However, the improvements w.r.t. the outlier removal are less significant in the full length experiments. Since for this dataset we used an up-sampling strategy that adds varying negative samples, the variation within these experiments is much higher, which makes the results less reliable and causes the observed differences to be less significant. Still, since the results broadly agree with the results for the more reliable TREC20 dataset, we take this as confirmation for the conclusions drawn there.

We also report the Outlierness metric, as introduced in [25], to show that the improvement of FELIX$_{iter=20}$ is not just due to the evaluation metric introduced in this paper but that there is an actual improvement w.r.t. the outlier use case.

We conclude that in our experiments, FELIX$_{iter=20}$ is able to effectively reduce the probability that a ranking with unknown exposure distribution is shown to the user, without a drop in utility or fairness, compared to other fair ranking methods and OMIT.

Discussion. If we compare our results to those in [25], OMIT does not perform as well as expected w.r.t. $P(u \mid \pi)$ and Outlierness. We see two reasons for this. First, OMIT considers outliers in the context of the whole list, while we consider outliers in the context of the top-k that they are presented in; their approach is able to remove outliers defined in the global context from the rankings but does not consider the outliers in the local context they are presented in, which is what we are evaluating for.

Second, in this paper we consider individual fairness, while Sarvi et al. [25] report results on group fairness. For individual fairness the number of constraints is much higher, therefore the space we are optimizing over is smaller, making it challenging for OMIT to find a good solution that is optimized for both utility and reducing outliers while satisfying all the fairness constraints. FELIX$_{iter=20}$ does not suffer from this, since, instead of adding an additional objective term to the optimization, it intervenes at the decomposition step, making it independent from the constraints used in the optimization.

This comparison shows that FELIX is very general in terms of use cases that it can be applied to. The condition that determines whether a ranking has a known exposure distribution can be focused on each individual ranking without having to rely on global assumptions. This allows us to really consider inter-item dependencies, while OMIT needs to work with the heuristic of global outliers instead. This also highlights the advantages of FELIX over the PL-ranker method. While for most experiments there was no significant difference in utility and fairness between those two methods, considering inter-item dependencies within the rankings is not possible for the PL approach to fair ranking.

7 SENSITIVITY ANALYSIS OF FELIX

Given the results obtained in the previous section, we now analyze the ability of FELIX to reduce the number of rankings with unknown exposure distribution along two important dimensions: (D1) the number of available item candidates; and (D2) the number of re-sampling iterations, $iter$ (see line 3 in Algorithm 2).

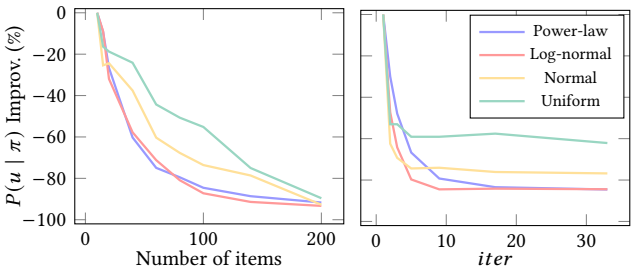


Figure 2: Sensitivity analysis. Relative reduction in $P(u \mid \pi)$ in % on the y-axis for different numbers of available candidate items (left) and different numbers of iterations (right).

For the TREC datasets most queries have less than 40 items, hence, we use a simulated set-up. This gives us more control, allowing us to observe FELIX's behaviour for different distributions and numbers of candidate items. Each analysis is conducted with a series of $m = 100$ simulated sets of n items (one can think of these item-sets as corresponding to m imaginary queries). Since we want to focus on the effectiveness of FELIX, rather than the quality of the predicted labels, we assume that for each item we know the correct probability that an item is relevant to users. For our analysis we sample these scores uniformly in the interval $[0, 1]$. The feature that is used for the outlier detection is sampled from a different probability distribution. We conduct experiments on the uniform, normal, log-normal, and power-law distribution to see how dependent the results are on the underlying data distribution. Each of the different distributions has a different base probability for a list of a given length to contain an outlier, and hence can be seen as different levels of difficulty for removing the rankings with unknown exposure distribution. With the definition of outliers used in this paper and a list length of 10, the probability that such a list contains an outlier is 0.6% for the uniform, 2.7% for the normal, 36.3% for the log normal and 60.5% for the power-law distribution.

(D1) Candidate items. The left plot in Fig. 2 shows the relative reduction of rankings with outliers with a varying number of candidate items. We use 20 re-sampling iterations. We see that for all distributions, FELIX performs increasingly better as the number of items increases. Having more items to shuffle between various rankings gives the method more flexibility in putting outlier items into different contexts, in which they do not appear as outliers.

(D2) Re-sampling parameter. The right plot of Fig. 2 shows how well FELIX is able to remove outliers from the rankings based on the number of re-sampling iterations, which is the only new hyper-parameter introduced by our method. We use 100 candidate items per query. We find that with an increasing number of re-samples, FELIX can remove more outliers. Nevertheless, the gains seem to be diminishing, depending on the distribution after 5–20 iterations.

Broader implications. Ranking systems often work in two stages, where in the first stage a certain number of documents are retrieved and in the second stage they are re-ranked with help of a learning to rank method. Our analysis of the number of candidate items (D1) can help deciding on how many items to retrieve in the first stage. Moreover, the analysis of the re-sampling parameter (D2) can help with deciding on a good performance/computation time trade-off when choosing the number of allowed re-sampling iterations.

8 CONCLUSION

Motivated by recent work on the impact of outliers on the exposure distribution within a ranking, we introduced the task of fair ranking under incomplete exposure estimation. We defined a new method, FELIX, that avoids showing rankings to the user which, due to inter-item dependencies, have unknown exposure distribution. We extended the convex optimization approach to fairness to the top-k setting and gave an efficient implementation of the algorithm that makes it feasible, even for a large number of items. We showed empirically that FELIX is able to significantly reduce the probability of generating rankings with unknown exposure, without hurting user utility or fairness compared to previous fair ranking methods.

FELIX is a first step towards fair ranking in cases where due to inter-item dependencies there is uncertainty about the exposure distribution of some rankings. By defining an efficient algorithm for the top-k setting, we enable the usage of the convex optimization approach towards fairness for use cases with a large number of items, which previously had been infeasible. We discussed that this approach gives more flexibility than other methods and allows, for example, to consider the relationship between items.

One limitation of our work is that, since the policy achieved by the convex optimization is only fair in expectation, this approach is most useful for head queries with a large number of repetitions. Use cases where this might be applied include job search, where next to the individual fairness criterion a correction for historical biases should be considered, or item search for items that are frequently bought. Second, our results are based on the assumption that the unknown exposure comes from inter-item dependencies and that the same items that cause one ranking to have unknown exposure distribution, when placed in another context will result in a ranking with known exposure distribution. This assumption holds for rankings with visible outliers, however, to prove the generalizability of this approach, experiments with other use cases are needed. Lastly, to have enough flexibility within the Birkhoff-von Neumann decomposition algorithm, enough entries of this matrix need to be non-zero. Using group fairness with only two groups, results in a marginal rank probability matrix that is a linear combination of just two permutation matrices [27]. More groups introduce more stochasticity, therefore this method is particularly interesting when working with individual fairness or a larger number of groups.

A potential direction for future work is to investigate whether FELIX can be extended for different user models. In this work we assume that most rankings follow a position-based exposure distribution. For other user-models like the cascade model a different approach might be necessary. Also, more research needs to be done on inter-item dependencies between items in a ranking and their impact on the exposure for different use cases. Phenomena like outliers or click bait have been explored to some extent but other types of cognitive bias that impact how we perceive items in relation to others have been broadly unexplored in the context of ranking systems. Lastly, extending user models to include inter-item dependencies such as outliers might allow for a more direct approach to fair ranking in cases where the exposure distribution is unknown.

DATA AND CODE

To facilitate reproducibility of our work, all code and parameters are shared at https://github.com/MariaHeuss/2022-SIGIR-FOE-Incomplete-Exposure.

ACKNOWLEDGEMENTS

We thank our reviewers for valuable feedback. This research was supported by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl, and by Ahold Delhaize. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

A PROOFS

A.1 Extended proof for the generalized Birkhoff-von Neumann

We give a more detailed proof of Lemma 4.2 and Theorem 4.1. Recall that we say that $P' \in \mathbb{R}^{n' \times k'}$ is an *extension* of $P \in \mathbb{R}^{n \times k}$ if $n' \geq n, k' \geq k$, and $P_{i,j} = P'_{i,j}$ for all (i, j) with $i \leq n$ and $j \leq k$. We denote this by $P'|_{i \leq n, j \leq k} = P$.

Lemma A.1. *Let $P = \{a_{i,j}\}_{i \leq n, j \leq k}$ be a matrix with the same properties as described in Theorem 4.1 with $k \leq n$. Then there is a matrix $P' = \{a'_{i,j}\}_{i \leq n, j \leq n}$ with $\forall i, j : 0 \leq a'_{i,j} \leq 1$ such that $P = P'|_{i \leq n, j \leq k}$, and $\forall i : \sum_{j=1}^{n} a'_{i,j} = 1$ and $\forall j : \sum_{i=1}^{n} a'_{i,j} = 1$.*

Proof. Define $P' = \{a'_{i,j}\}_{i \leq n, j \leq n}$ as

$$a'_{i,j} = \begin{cases} a_{i,j} & \text{if } j \leq k \\ \frac{1 - \sum_{j'=1}^{k} a_{i,j'}}{n-k} & \text{if } j > k. \end{cases} \tag{12}$$

Then $P'|_{i \leq n, j \leq k} = P$ by definition. Since for all i, $0 \leq \sum_{j=1}^{k} a_{i,j} \leq 1$ we also have $0 \leq \frac{1 - \sum_{j'=1}^{k} a_{i,j'}}{n-k} \leq 1$. Moreover, for all $i \leq n$:

$$\begin{aligned} \sum_{j=1}^{n} a'_{i,j} &= \sum_{j=1}^{k} a_{i,j} + \sum_{j=k+1}^{n} \frac{1 - \sum_{j'=1} a_{i,j'}}{n-k} \\ &= \sum_{j=1}^{k} a_{i,j} + (n-k) \cdot \frac{1 - \sum_{j'=1}^{k} a_{i,j'}}{n-k} \\ &= \sum_{j=1}^{k} a_{i,j} + (1 - \sum_{j'=1}^{k} a_{i,j'}) \\ &= 1, \end{aligned}$$

where we used in the second equality that we sum over $(n-k)$ times the same value. We know that the columns of the matrix sum to 1 for all $j \leq k$, since this is the case for P. For $j > k$ we have:

$$\begin{aligned} \sum_{i=1}^{n} a'_{i,j} &= \frac{1}{n-k} (\sum_{j=k}^{n} \sum_{i=1}^{n} a'_{i,j}) \\ &= \frac{1}{n-k} (n - \sum_{j=1}^{k} \sum_{i=1}^{n} a'_{i,j}) \\ &= \frac{n-k}{n-k} = 1. \end{aligned}$$

Here in the first equality we used that all columns from the k-th column are the same. In the second equality we used that since all rows are summing to 1, the sum of all rows (and therefore also the sum of all columns) equals n. The last equality simply uses the fact that each of the first k columns sums to 1. □

We use this Lemma to prove the generalized Birkhoff-von Neumann theorem. Let $k \leq n$.

Theorem A.2. *Any matrix $P = \{a_{i,j}\}_{i \leq n, j \leq k}$ with $\forall i, j : 0 \leq a_{i,j} \leq 1$, $\forall j : \sum_{i=1}^{n} a_{i,j} = 1$ and $\forall i : \sum_{j=1}^{k} a_{i,j} \leq 1$ can be written as the convex sum $P = \sum_{l=1}^{m} \alpha_l \cdot P_l$ of permutation matrices P_l with coefficients $\alpha_l \in [0, 1]$ such that $\sum_{l=1}^{m} \alpha_l = 1$.*

Proof. In Lemma A.1 we show that P can be extended to a doubly stochastic matrix P', i.e. $P = P'|_{i \leq n, j \leq k}$. For this matrix P', the theorem by Birkhoff and von Neumann states that we can find a decomposition into the convex sum of permutation matrices, $P' = \sum_{l=1}^{m} \alpha_l P'_l$, with $\alpha_l \in [0, 1]$, $\sum_{l=1}^{m} \alpha_l = 1$ and P'_l permutation matrices. This induces a decomposition of the original matrix P:

$$P = \sum_{l=1}^{m} \alpha_l P'_l|_{i \leq n, j \leq k}. \qquad \square$$

A.2 Complexity of the generalized Birkhoff-von Neumann algorithm

In this section we prove the following claim from Section 4.2:

Theorem A.3. *Using the modified top-k algorithm for the generalized Birkhoff-von Neumann theorem, Algorithm 1, a decomposition as described in Theorem 4.1 can be obtained with time complexity $O(k^3 n^2)$.*

Proof. The time complexity of Algorithm 1 depends on the complexity of the adjusted Hopcroft-Karp algorithm (line 5) and the number of times it needs to be executed (line 4–9), which is equal to the number of permutation matrices in the decomposition. Hopcroft and Karp [18] show that the time complexity of the Hopcroft-Karp algorithm is $O((m+l)\sqrt{l})$, where l is the number of vertices and m is the number of edges in the biparate graph. For the baseline approach we have $l = 2n$ and $m = n^2$, therefore the complexity of the Hopcroft-Karp algorithm in this setting would be $O(n^2\sqrt{(n)})$. Using our approach instead, we have $l = n + (k+1)$ and $m = n \cdot (k+1)$ which reduces the complexity to $O(kn\sqrt{(n)})$. Furthermore since the maximum length of each augmenting path is bounded by $2 \cdot k$, we can substitute the $\sqrt{n}$ term with k (see Corollary 2 and Theorem 3 of [18]). This gives us a time complexity of $O(k^2 n)$ for the full matching algorithm. For the number of matrices in the decomposition, Johnson et al. [21] define an upper bound of $n^2 - 2n + 2$ permutation matrices, which means that the total complexity of the Birkhoff-von Neumann algorithm equals $O(n^4\sqrt{n})$. Since for our algorithm, a loose upper bound for the number of permutation matrices is $k \cdot n$, the algorithm proposed in this paper has a time complexity of only $O(n^2 k^3)$, which makes it much more feasible than the more naive algorithm proposed in Section 4.1 for large values of n. □

REFERENCES

[1] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating Position Bias without Intrusive Intervunentions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 474–482.

[2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 385–394.

[3] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of Attention: Amortizing Individual Fairness in Rankings. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 405–414.

[4] Garrett Birkhoff. 1940. *Lattice Theory*. AMS.

[5] Garrett Birkhoff. 1946. Tres observaciones sobre el algebra lineal. *Univ. Nac. Tucuman, Ser. A* 5 (1946), 147–154.

[6] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[7] Carlos Castillo. 2019. Fairness and Transparency in Ranking. In *ACM SIGIR Forum*, Vol. 52. 64–71.

[8] 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)*. 28:1–28:15.

[9] Yuxin Chen and Changho Suh. 2015. Top-K Ranking: An Information-theoretic Perspective. In *2015 IEEE Information Theory Workshop - Fall (ITW)*. 212–213.

[10] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. 2015. *Click Models for Web Search*. Morgan & Claypool Publishers.

[11] Prasad M Deshpande, Deepak P, and Krishna Kummamuru. 2008. Efficient Online Top-K Retrieval with Arbitrary Similarity Measures. In *Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology*. 356–367.

[12] Fernando Diaz, Bhaskar Mitra, Michael D Ekstrand, Asia J Biega, and Ben Carterette. 2020. Evaluating Stochastic Rankings with Expected Exposure. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 275–284.

[13] Fanny Dufossé, Kamer Kaya, Ioannis Panagiotas, and Bora Uçar. 2018. Further notes on Birkhoff–von Neumann decomposition of doubly stochastic matrices. *Linear Algebra Appl.* 554 (2018), 68–78.

[14] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2021. Fairness and Discrimination in Information Access Systems. *arXiv preprint arXiv:2105.05779* (2021).

[15] Michael D. Ekstrand, Graham McDonald, Amifa Raj, and Isaac Johnson. 2022. Overview of the TREC 2021 Fair Ranking Track. In *The Thirtieth Text REtrieval Conference (TREC 2021) Proceedings*.

[16] Zhichong Fang, Aman Agarwal, and Thorsten Joachims. 2019. Intervention Harvesting for Context-Dependent Examination-Bias Estimation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 825–834.

[17] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-aware Ranking in Search & Recommendation Systems with Application to Linkedin Talent Search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2221–2231.

[18] John E. Hopcroft and Richard M. Karp. 1973. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM J. Comput.* 2, 4 (1973), 225–231.

[19] Thorsten Joachims. 2021. Fairness and Control of Exposure in Two-sided Markets. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*.

[20] 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.

[21] Diane M Johnson, AL Dulmage, and NS Mendelsohn. 1960. On an algorithm of G. Birkhoff concerning doubly stochastic matrices. *Canad. Math. Bull.* 3, 3 (1960), 237–242.

[22] 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 *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2243–2251.

[23] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* (2009), 333–389.

[24] Piotr Sapiezynski, Wesley Zeng, Ronald E Robertson, Alan Mislove, and Christo Wilson. 2019. Quantifying the Impact of User Attention on Fair Group Representation in Ranked Lists. In *Companion Proceedings of The 2019 World Wide Web Conference*. 553–562.

[25] Fatemeh Sarvi, Maria Heuss, Mohammad Aliannejadi, Sebastian Schelter, and Maarten de Rijke. 2022. Understanding and Mitigating the Effect of Outliers in Fair Ranking. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 861–869.

[26] Ashudeep Singh and Thorsten Joachims. 2017. Equality of Opportunity in Rankings. In *Workshop on Prioritizing Online Content (WPOC) at NIPS*. 31.

[27] 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.

[28] Ashudeep Singh and Thorsten Joachims. 2019. Policy Learning for Fairness in Ranking. In *Advances in Neural Information Processing Systems*, Vol. 32.

[29] Ashudeep Singh, David Kempe, and Thorsten Joachims. 2021. Fairness in Ranking under Uncertainty. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 11896–11908.

[30] Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language Model Information Retrieval with Document Expansion. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. 407–414.

[31] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2021. Mixture-Based Correction for Position and Trust Bias in Counterfactual Learning to Rank. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1869–1878.

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

[33] Ali Vardasbi, Fatemeh Sarvi, and Maarten de Rijke. 2022. Probabilistic Permutation Graph Search: Black-Box Optimization for Fairness in Ranking. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

[34] Lequn Wang and Thorsten Joachims. 2021. User Fairness, Item Fairness, and Diversity for Rankings in Two-Sided Markets. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 23–41.

[35] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1288–1297.

[36] 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.

[37] 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.

[38] Himank Yadav, Zhengxiao Du, and Thorsten Joachims. 2021. Policy-Gradient Training of Fair and Unbiased Ranking Functions. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1044–1053.

[39] Ke Yang and Julia Stoyanovich. 2017. Measuring Fairness in Ranked Outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. Article 22, 6 pages.

[40] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. 2012. On Top-k Recommendation Using Social Networks. In *Proceedings of the Sixth ACM Conference on Recommender Systems*. 67–74.

[41] 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 International Conference on World Wide Web*. 1011–1018.

[42] 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.

[43] 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.

[44] Meike Zehlike, Tom Sühr, Ricardo Baeza-Yates, Francesco Bonchi, Carlos Castillo, and Sara Hajian. 2022. Fair Top-k Ranking with multiple protected groups. *Information Processing & Management* 59, 1 (2022), 102707.

[45] Meike Zehlike, Ke Yang, and Julia Stoyanovich. 2021. Fairness in Ranking: A Survey. *arXiv preprint arXiv:2103.14000* (2021).

[46] Chengxiang Zhai and John Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 334–342.