Fast online ranking with fairness of exposure

Nicolas Usunier
usunier@fb.com
Meta AI
Paris, France

Virginie Do
virginiedo@fb.com
Meta AI
Paris, France

Elvis Dohmatob
dohmatob@fb.com
Meta AI
Paris, France

ABSTRACT

As recommender systems become increasingly central for sorting and prioritizing the content available online, they have a growing impact on the opportunities or revenue of their items producers. For instance, they influence which recruiter a resume is recommended to, or to whom and how much a music track, video or news article is being exposed. This calls for recommendation approaches that not only maximize (a proxy of) user satisfaction, but also consider some notion of fairness in the exposure of items or groups of items. Formally, such recommendations are usually obtained by maximizing a concave objective function in the space of randomized rankings. When the total exposure of an item is defined as the sum of its exposure over users, the optimal rankings of every users become coupled, which makes the optimization process challenging. Existing approaches to find these rankings either solve the global optimization problem in a *batch* setting, i.e., for all users at once, which makes them inapplicable at scale, or are based on heuristics that have weak theoretical guarantees. In this paper, we propose the first efficient *online* algorithm to optimize concave objective functions in the space of rankings which applies to every concave and smooth objective function, such as the ones found for fairness of exposure. Based on online variants of the Frank-Wolfe algorithm, we show that our algorithm is *computationally fast*, generating rankings on-the-fly with computation cost dominated by the sort operation, *memory efficient*, and has *strong theoretical guarantees*. Compared to baseline policies that only maximize user-side performance, our algorithm allows to incorporate complex fairness of exposure criteria in the recommendations with negligible computational overhead. We present experiments on artificial music and movie recommendation tasks using Last.fm and MovieLens datasets which suggest that in practice, the algorithm rapidly reaches good performances on three different objectives representing different fairness of exposure criteria.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

fairness, recommender systems, online ranking

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.

FAccT '22, June 21–24, 2022, Seoul,Republic of Korea

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9352-2/22/06…$15.00
https://doi.org/10.1145/3531146.3534633

ACM Reference Format:
Nicolas Usunier, Virginie Do, and Elvis Dohmatob. 2022. Fast online ranking with fairness of exposure. In *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT '22), June 21–24, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3531146.3534633

1 INTRODUCTION

Recommender systems are ubiquitous in our lives, from the prioritization of content in news feeds to matching algorithms for dating or hiring. The objective of recommender systems is traditionally formulated as maximizing a proxy for user satisfaction such as ranking performance. However, it has been observed that these recommendation strategies can have undesirable side effects. For instance, several authors discussed popularity biases and winner-take-all effects that may lead to disproportionately expose a few items even if they are assessed as only slightly better than others [1, 3, 34], or disparities in content recommendation across social groups defined by sensitive attributes [18, 37]. An approach to mitigate these undesirable effects is to take a more general perspective to the objective of recommendation systems. Considering recommendation as an allocation problem [33, 34] in which the "resource" is the exposure to users, the objective of recommender systems is to allocate this resource fairly, i.e., by taking into account the interests of the various stakeholders – users, content producers, social groups defined by sensitive attributes – depending on the application context. This perspective yields the traditional objective of recommendation when only the ranking performance averaged over individual users is taken into account.

There are two main challenges associated with the fair allocation of exposure in recommender systems. The first challenge is the specification of the formal objective function that defines the trade-off between the possibly competing interests of the stakeholders in a given context. The second challenge is the design of a scalable algorithmic solution: when considering the exposure of items across users in the objective function, the system needs to account for what was previously recommended (and, potentially, to whom) when generating the recommendations for a user. This requires solving a global optimization problem in the space of the rankings of all users. In contrast, traditional recommender systems simply sort items by estimated relevance to the user, irrespective of what was recommended to other users.

In this paper, we address the algorithmic challenge, with a solution that is sufficiently general to capture many objective functions for ranking with fairness of exposure, leaving the choice of the exact objective function to the practitioner. Following previous work on fairness of exposure, we consider objective functions that are concave functions that should be optimized in the space of randomized rankings [11, 32, 34, 35]. Our algorithm, Offr (Online Frank-Wolfe for Fair Ranking), is a computationally efficient algorithm that optimizes such objective functions *online*, i.e., by generating rankings on-the-fly as users request recommendations. The algorithm dynamically modifies item scores to optimize for both user utility and the selected fairness of exposure objective. We prove that the objective function converges to the optimum in $O(1/\sqrt{t})$, where t is the number of time steps. The computational complexity of Offr at each time step is dominated by the cost of sorting, and it requires only $O(\#users + \#items)$ storage. The computation cost of Offr are thus of the same order as what is required in traditional recommenders systems. Consequently, using Offr, taking into account fairness of exposure in the recommendations is (almost) free. Our main technical insight is to observe that in the context of fair ranking, the usage of Frank-Wolfe algorithms [12] resolves two difficulties:

(1) Frank-Wolfe algorithms optimize in the space of probability distributions but use at each round a deterministic outcome as the update direction. In our case, it means that Offr outputs a (deterministic) ranking at each time step while implicitly optimizing in the space of randomized rankings.
(2) Even though the space of rankings is combinatorial, the objective functions used in fairness of exposure have a linear structure that Frank-Wolfe algorithms can leverage, as already noticed by Do et al. [11].

Compared to existing algorithms, Offr is the first widely applicable and scalable algorithm for fairness of exposure in rankings. Existing online ranking algorithms for fairness of exposure [3, 32, 41] are limited in scope as they apply to only a few possible fairness objectives, and only have weak theoretical guarantees. Do et al. [11] show how to apply the Frank-Wolfe algorithm to general smooth and concave objective functions for ranking. However, they only solve the problem in a *batch* setting, i.e., computing the recommendations of all users at once, which makes the algorithm impractical for large problems, because of both computation and memory costs. Our algorithm can be seen as an online variant of this algorithm, which resolves all scalability issues.

We showcase the generality of Offr on three running examples of objective functions for fairness of exposure. The first two objectives are welfare functions for two-sided fairness [11], and the criterion of quality-weighted exposure [3, 34]. The third objective, which we call *balanced exposure to user groups*, is novel. Taking inspiration from audits of job advertisement platforms [18], this objective considers maximizing ranking performance while ensuring that each item is evenly exposed to different user groups defined by sensitive attributes.

In the remainder of the paper, we present the recommendation framework and the different fairness objectives we consider in the next section. In Sec. 3, we present our online algorithm in its most general form, as well as its regret bound. In Sec. 4, we instantiate the algorithm on three fairness objectives and provide explicit convergence rates in each case. We present our experiments in Sec. 5. We discuss the related work in Sec. 6. Finally, in Sec. 7, we discuss the limitations of this work and avenues for future research.

2 FAIRNESS OF EXPOSURE IN RANKINGS

This paper addresses the online ranking problem, where users arrive one at a time and the recommender system produces a ranking of k items for that user. We focus on an abstract framework where the recommender system has two informal goals. First, the recommended items should be relevant to the user. Second, the exposure of items should be distributed "fairly" across users, for some definition of fairness which depends on the application context. We formalize these two goals in this section, by defining objective functions composed of a weighted sum of two terms: the *user objective* which depends on the ranking performance from the user's perspective, and the *fairness objective*, which depends on the exposure of items. In this section, we focus on the *ideal* objective functions, which are defined in a *static* ranking framework. In the next sections, we focus on the online ranking setting, where at each time step, an incoming user requests recommendations and the recommender systems produces the recommendation list on-the-fly while optimizing these ideal objective functions.

In order to disentangle the problem of learning user preferences from the problem of generating fair recommendations, we consider that user preferences are given by an oracle. We start this section by describing the recommendation framework we consider. We then present the fairness objectives we focus on throughout the paper.

Notation. Integer intervals are denoted within brackets, i.e., $\forall n \in \mathbb{N}$, $[\![n]\!] = \{1, .., n\}$. We use the Dirac notation $\langle x|y\rangle$ for the dot product of two vectors of same dimension x and y. Finally, $\mathbb{1}_{\{\text{expr}\}}$ is 1 when expr is true, and 0 otherwise.

2.1 Recommendation framework

We consider a recommendation problem with n users and m items. We identify the set of users with $[\![n]\!]$ and the set of items with $[\![m]\!]$. We denote by $\mu_{ij} \in [0, 1]$ the value of recommending item j to user i (e.g., a rating normalized in $[0, 1]$). To account for the fact that users are more or less frequent users of the platform, we define the *activity* of user i as a weight $w_i \in [0, 1]$. We consider that $w = (w_1, ..., w_n)$ is a probability distribution, so that in the online setting described later in this paper, w_i is the probability that the current user at a given time step is i.

The recommendation for a user is a *top-k ranking* (or simply *ranking* when the context is clear), i.e., a sorted list of k unique items, where typically $k \ll m$. Formally, we represent a ranking by a mapping $\sigma : [\![k]\!] \to [\![m]\!]$ from ranks to recommended items with the constraint that different ranks correspond to different items. The ranking performance on the user side follows the *position-based* model, similarly to previous work [3, 11, 33–35]. Given a set of non-negative, non-increasing *exposure weights* $b = (b_1, ..., b_k)$, the ranking performance of σ for user i, denoted by $u_i(\sigma)$, is equal to:

$$u_i(\sigma) = \sum_{r=1}^{k} \mu_{i,\sigma(r)} b_r \qquad \text{with } b_1 \geq \ldots \geq b_k \geq 0. \tag{1}$$

We use the shorthand *user utility* to refer to u_i. Following previous work on fairness of exposure, we interpret the weights in b as being commensurable to the exposure an item receives given its rank. The weights are non-increasing to account for the *position bias*, which means that the user attention to an item decreases with the rank of the item. Given a top-k ranking σ, the *exposure vector induced by* σ, denoted by $E(\sigma) \in \mathbb{R}^m$ assigns each item to its exposure in σ:

$$\forall j \in [\![m]\!], E_j(\sigma) = \begin{cases} b_r & \text{if } \exists r \in [\![k]\!], \sigma(r) = j \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

The user utility is then equal to $u_i(\sigma) = \sum_{j=1}^{m} \mu_{ij} E_j(\sigma) = \langle \mu_i | E(\sigma) \rangle$.

In practice, the ranking given by a recommender system to a user is not necessarily unique: previous work in *static* rankings consider randomization in their rankings [34], while in our case of online ranking, it is possible that the same user receives different rankings at different time steps. In that case, we are interested in averages of user utilities and item exposures. To formally define these averages, we use the notation:

$$\begin{aligned} &\mathcal{E} = \{E(\sigma) : \sigma \text{ is a top-}k \text{ ranking}\} \\ &\overline{\mathcal{E}} = \text{convexhull}(\mathcal{E}) \qquad \Pi = \overline{\mathcal{E}}^n. \end{aligned} \tag{3}$$

$\mathcal{E}$ is the set of possible item exposures vectors and $\overline{\mathcal{E}}$ is the set of possible average exposure vectors. Π is an *exposure matrix*, where π_{ij} is the average exposure of item j to user i. Under the position-based model, a matrix $\pi \in \Pi$ characterizes a recommender system since it specifies the average exposure of every item to every user. We use π as a convenient mathematical device to study the optimization problems of interests, keeping in mind that out algorithms effectively produce a ranking at each time step.

Recalling that w represents the user activities, the user utilities and total item exposures under π are defined as

$$\begin{aligned} &\text{(utility of user } i) \qquad u_i(\pi) = \langle \mu_i | \pi_i \rangle \\ &\text{(exposure of item } j) \quad v_j(\pi) = \sum_{i=1}^{n} w_i \pi_{ij}. \end{aligned} \tag{4}$$

Fairness of exposure refers to objectives in recommender systems where maximizing average user utility is not the sole or main objective of the system. Typically, the exposure of items v_j, or variants of them, should also be taken into account in the recommendation. We formulate the goal of a recommender system as optimizing an objective function $f(\pi)$ over $\pi \in \Pi$, where f accounts for both the user utility and the fairness objectives.

2.2 Fairness Objectives

We now present our three examples of objective functions $f(\pi)$ in order of "difficulty" to perform online ranking compared to static ranking. In all three cases, it is easy to see that the objective functions are *concave* with respect to the recommended exposures π. The objective functions should be maximized, so the optimal exposures π^* satisfy

$$\pi^* \in \underset{\pi \in \Pi}{\text{argmax}}\, f(\pi). \tag{5}$$

Since our algorithm works on any concave function of the average exposures respecting some regularity conditions, we emphasize that the three objective functions below are only a few examples among many.

Two-sided fairness. The first example is from Do et al. [11] who optimize an additive concave welfare function of user utilities and item exposures. Interpreting item exposure as the utility of the item's producer, this approach is grounded into notions of distributive justice from welfare economics and captures both user- and item-fairness [11]. For $\eta > 0$, $\beta > 0$ and $\alpha_1 \in (-\infty, 1), \alpha_2 \in (-\infty, 1)$, the objective function is:

$$\begin{aligned} &f(\pi) = \sum_{i=1}^{n} w_i \psi_{\alpha_1}\big(u_i(\pi)\big) + \frac{\beta}{m} \sum_{j=1}^{m} \psi_{\alpha_2}\big(v_j(\pi)\big) \\ &\text{where } \psi_\alpha(x) = \begin{cases} \text{sign}(\alpha)(\eta + x)^\alpha & \text{if } \alpha \neq 0 \\ \log(\eta + x) & \text{if } \alpha = 0 \end{cases}. \end{aligned} \tag{6}$$

Where $\eta > 0$ avoids infinite derivatives at 0, $\beta > 0$ controls the relative weight of user-side and item-side objectives, and $\alpha_1 < 1$ (resp. $\alpha_2 < 1$) controls how much we focus on maximizing the utility of the worse-off users (resp. items) [11].

Quality-weighted exposure. One of the main criteria for fairness of exposure is *quality-weighted* exposure [3, 40] (also called merit-based fairness [32, 34]). A measure q_j of the overall quality of an item is taken as reference, and the criterion stipulates that the item exposure is proportional to its quality. q_j is often defined as the average value μ_{ij} over users. Using this definition of q_j, as noted by Do et al. [11], it is possible to optimize trade-offs between average user utility and proportional exposure using a penalized objective of the form:

$$\begin{aligned} &f(\pi) = \sum_{i=1}^{n} w_i u_i(\pi) - \beta \sqrt{\eta + \frac{1}{m} \sum_{j=1}^{m} \Big(q_{\text{avg}} v_j(\pi) - q_j \|b\|_1\Big)^2} \\ &\text{where } q_j = \sum_{i=1}^{n} w_i \mu_{ij} \text{ and } q_{\text{avg}} = \frac{1}{m} \sum_{j=1}^{m} q_j. \end{aligned} \tag{7}$$

As before, $\beta > 0$ controls the trade-off between user utilities and the fairness of exposure penalty and $\eta > 0$ avoids infinite derivatives at 0. This form of the exposure penalty was chosen because it is concave and differentiable, and it is equal to zero when exposure is exactly proportional to quality, i.e., when $\forall j, j', \frac{v_j}{q_j} = \frac{v_{j'}}{q_{j'}}$. We use $q_{\text{avg}} v_j(\pi) - q_j \|b\|_1$ rather than $\frac{v_j(\pi)}{q_j} - \frac{\|b\|_1}{q_{\text{avg}}}$ because the the former is more stable when qualities are close to 0 or estimated.

Balanced exposure to user groups. We also propose to study a new criterion we call *balanced exposure to user groups*, which aims at exposing every item evenly across different user groups. For instance, a designer of a recommendation system might want to ensure a job ad is exposed to the similar proportion of men and women [18], or to even proportions within each age category. Let $\mathcal{S} = (s_1, ..., s_{|\mathcal{S}|})$ be a set of non-empty groups of users. We do not need $\mathcal{S}$ to contain all users, and groups may be overlapping. Let $v_{j|s}$ be the exposure of item j within the group s, i.e., the amount of exposure j receives in group s with respect to the total exposure available for this group. That is, for any $\pi \in \Pi$, define

$$v_{j|s}(\pi) := \sum_{i \in s} \frac{w_i}{\overline{w}_s} \pi_{ij}, \text{ with } \overline{w}_s := \sum_{i \in s} w_i. \quad v_{j|\text{avg}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} v_{j|s}(\pi)$$

Also, let $v_{j|\text{avg}} := (1/|\mathcal{S}|) \sum_{s \in \mathcal{S}} v_{j|s}(\pi)$ be the average exposure for item j, across all the groups. The objective function we consider takes the following form, where $\beta > 0$ and $\eta > 0$ play the same roles as before:

$$f(\pi) = \sum_{i=1}^{n} w_i u_i(\pi) - \frac{\beta}{m} \sum_{j=1}^{m} \sqrt{\eta + \sum_{s \in \mathcal{S}} \left(v_{j|s}(\pi) - v_{j|\text{avg}}(\pi) \right)^2}. \tag{8}$$

3 FAST ONLINE RANKING

3.1 Online ranking

The *online setting* we consider is summarized as follows. At each time step $t \geq 1$:

(1) A user $i^{(t)} \in [\![n]\!]$ asks for recommendations. We assume $i^{(t)}$ is drawn at random from the fixed but unknown distribution of user activities with parameters w, i.e., $i^{(t)} \sim$ Categorical(w).
(2) The recommender system picks a ranking $\sigma^{(t)}$.

Note that as stated before, the main assumptions of this framework are the fact that incoming users are sampled independently at each step from a distribution that remains constant over time. In our setting, we consider that the (user, item) values μ_{ij} are known to the system. However, the user activities w_i are unknown.

Let $e^{(t)} = E(\sigma^{(t)})$ be the exposure vector induced by $\sigma^{(t)}$, and define, for every user i:

- The user counts at time t: $c_i^{(t)} = \sum_{\tau \leq t} \mathbb{1}_{\{i^{(\tau)}=i\}}$;
- The average exposure at time t: $\pi_i^{(t)} = \frac{1}{c_i^{(t)}} \sum_{\tau \leq t} \mathbb{1}_{\{i^{(\tau)}=i\}} e^{(\tau)}$.

Given an objective function f such as the ones defined in the previous section, our goal is to design computationally efficient algorithms with low *regret* when t grows to infinity. More formally the goal of the algorithm is to guarantee:

$$R^{(t)} = \max_{\pi \in \Pi} \left[f(\pi) \right] - \mathbb{E}[f(\pi^{(t)})] \xrightarrow[t \to \infty]{} 0 \tag{9}$$

where the expectation in $R^{(t)}$ is taken over the random draws of $i^{(1)}, \ldots,$ and the $O(.)$ hides constants that depend on the problem, such as the number of users or items.

3.2 The Offr algorithm

We describe in this section our generic algorithm, called Offr for Online Frank-Wolfe for Fair Ranking. Offr works with an abstract objective function $f_w : \Pi \to \mathbb{R}$, which is parameterized by the vector of user activities w. The $f_w(\pi)$ of this section is exactly the $f(\pi)$ of the previous section, except that we make explicit the dependency on w because it plays an important role in the algorithm and its analysis.

Assumptions on w and f_w. In the remainder, we assume w is fixed and non-degenerate, i.e., $\forall i \in [\![n]\!], w_i > 0$. We assume that for every w, $\pi \mapsto f_w(\pi)$ is *concave* and *differentiable*. More importantly, the fundamental object in our algorithm are the partial derivatives of f with respect to π_i normalized by w_i. Given a user index i, let

$$g_{w,i}(\pi) = \frac{1}{w_i} \frac{\partial f_w}{\partial \pi_i}(\pi) \in \mathbb{R}^m. \tag{10}$$

We assume that $g_{w,i}$ is bounded and Lipschitz with respect to π_i: for every $\pi \in \overline{\mathcal{E}}^n$ and every $\pi_i' \in \overline{\mathcal{E}}$, we have:

- Bounded gradients: $\|g_{w,i}(\pi)\|_\infty \leq G_i$;
- Lipschitz gradients:

$$\left\| g_{w,i}(\pi) - g_{w,i}(\pi_1, \ldots, \pi_{i-1}, \pi_i', \pi_{i+1}, \ldots, \pi_n) \right\|_2 \leq L_i \|\pi_i - \pi_i'\|_2.$$

Notice that with the normalization by w_i in $g_{w,i}$, these assumptions guarantee that the importance of a user is commensurable with their activity, i.e., that the objective does not depend disproportionately on users we never see.

Online Frank-Wolfe with an approximate gradient. Our algorithm is described by the following rule for choosing $e^{(t)}$. First, we rely on $\hat{g}_i^{(t)}$ which is an approximation of the gradient $g_{w,i}(\pi^{(t)})$. We describe later the required properties of the approximation we need is given (see Th. 2 below), and the approximation one we use in practice (see (14) below). Notice that we rely on an approximation because the user activities are unknown. Then, choose $e^{(t)}$ as:

$$e^{(t)} \in \operatorname*{argmax}_{e \in \mathcal{E}} \langle \hat{g}_{i^{(t)}}^{(t)} | e \rangle. \tag{11}$$

Since we compute a maximum dot product with a gradient (or an approximation thereof), our algorithm is a variant of online Frank-Wolfe algorithms. We discuss in more details the relationship with this literature in Sec. 6.

Frank-Wolfe algorithms shine when the argmax in (11) can be computed efficiently. As previously noted by Do et al. [11] who only study *static* ranking, Frank-Wolfe algorithms are particularly suited to ranking because (11) only requires a top-k sorting. Let topk(x) be a routine that returns the indices of k largest elements in vector x.[1] We have:

Proposition 1. *[11, Thm. 1]*

$$\sigma^{(t)} = \text{topk}\big(\hat{g}_i^{(t)}\big) \implies E\big(\sigma^{(t)}\big) \in \operatorname*{argmax}_{e \in \mathcal{E}} \langle \hat{g}_i^{(t)} | e \rangle.$$

We call Offr (Online Frank-Wolfe for Fair Ranking) the usage of the online Frank-Wolfe update (11) in ranking tasks, i.e., using Prop. 1 to efficiently perform the argmax computation of (11).

We are now ready to state our main result regarding the convergence of *dynamic* ranking. The result does not rely on the specific structure of ranking problems. The result below is valid as long as $\mathcal{E} \subset \mathbb{R}^m$ is a finite set with $\forall e \in \mathcal{E}, 0 \leq e_j \leq 1$. We denote by $B_{\mathcal{E}} = \max_{e \in \mathcal{E}} \|e\|_1$ ($B_{\mathcal{E}} = \|b\|_1$ in our case).

Theorem 2 (Convergence of (11)). *Let $\pi_0 \in \overline{\mathcal{E}}^n$, and assume there exists D_i such that $\forall t \geq 1$ and $\forall i \in [\![n]\!]$, we have:*

$$\mathbb{E}\left[\left\| \hat{g}_i^{(t)} - g_{w,i}\big(\pi^{(t-1)}\big) \right\|_\infty \right] \leq \frac{D_i}{\sqrt{t}}, \tag{12}$$

where the expectation is taken over $i^{(1)}, \ldots, i^{(t-1)}$.

Then, with $e^{(t)}$ chosen by (11) at all time steps, we have $\forall t \geq 1$:

$$R^{(t)} \leq 2B_{\mathcal{E}} \sum_{i=1}^{n} (L_i + G_i) \frac{\ln(et)}{t} + \frac{6B_{\mathcal{E}} \sum_{i=1}^{n} \sqrt{w_i}(G_i + D_i)}{\sqrt{t}} \tag{13}$$

[1] Formally, $\sigma = \text{topk}(x) \implies \Big(x_{\sigma(1)} \geq \ldots \geq x_{\sigma(k)}$ and $\forall j \notin \{\sigma(1), \ldots, \sigma(K)\}, x_j \leq x_{\sigma(k)} \Big)$, using an arbitrary tie breaking rule as it does not play any role in the analysis.

Appendix A is devoted to the proof of this result. The main technical difficulty comes from the fact that we only update the parameters of the incoming user $i^{(t)}$ with possibly non-uniform user activities, and we need a stochastic step size $1/c_i^{(t)}$ so that the iterates of the optimization algorithm match the resulting average exposures. Notice that the guarantee does not depend on the choice of π_0 because it only affects the first gradient computed for the user. In practice we set π_0 to the average exposure profile of a random top-k ranking.

Since we do not have access to the exact gradient because user activities are unknown, we use in practice the approximate gradient built using the empirical user activities:

$$\hat{g}_i^{(t)} = g_{\hat{w}^{(t-1)},i}\big(\pi^{(t-1)}\big) \qquad \text{where } \hat{w}^{(t)} = \frac{c^{(t)}}{t} \tag{14}$$

with a fallback formula when $\hat{w}_i^{(t-1)} = 0$. In the next section, we discuss the computationally efficient implementation of this rule for the three objectives of Sec. 2.2, and we provide explicit bounds for D_i of (12) in each case.

4 APPLICATIONS OF OFFR

Practical implementations of Offr do not rely on naive computations of $\hat{g}_i^{(t)} = g_{\hat{w}^{(t-1)},i}\big(\pi^{(t-1)}\big)$, because they would require explicitly keeping track of $\pi^{(t)}$. $\pi^{(t)}$ is a matrix of size $n \times m$, which is impossible to store explicitly in large-scale applications.[2] Importantly, as we illustrate in this section, for the objectives of Sec. 2.2 it is unnecessary to maintain explicit representations of $\pi^{(t)}$ because the gradients depend on $\pi^{(t)}$ only through utilities or exposures, for which we can maintain online estimates.

4.1 Practical implementations

The implementation of Offr for the three fairness objectives (6), (7) and (8) are described in Alg. 1, 2 and 3 respectively, where we dropped the superscripts $^{(t-1)}$ and $^{(t)}$ for better readability. At every round t, there are three steps:

(1) compute approximate gradients based on online estimates of user values and exposures,
(2) update the relevant online estimates of user utility and item exposures,
(3) perform a top-k sort of the scores computed in step (1) to obtain $\sigma^{(t)}$.

We omit the details of the calculation of $\hat{g}_{ij}$ in Alg. 1, 2 and 3, which are obtained by differentiation of $f_{\hat{w}}$ using (14).[3]

Two-sided Fairness. For two-sided fairness (6), we have:

$$g_{w,ij}\big(\pi^{(t)}\big) = \psi'_{\alpha_1}\big(u_i(\pi^{(t)}\big)\mu_i + \frac{\beta}{m}\psi'_{\alpha_2}\big(v_j(\pi^{(t)})\big). \tag{15}$$

[2] $n \times m$ is also the size of the matrix of (user, item) values μ, which in practice is not stored explicitly. Rather, the values μ_{ij} are computed on-the-fly (possibly using caching for often-accessed values) and the storage uses compact representations, such as latent factor models [23] or neural networks [16].

[3] In Alg. 3, we use a factor $\frac{t}{c_{s[i]}+1}$ while the direct calculation would give a factor $\frac{t}{c_{s[i]}}$. The formula we use more gracefully deals with the case $c_{s[i]} = 0$ and enjoys similar bounds when t is large.

Let:

$$\begin{aligned} &\forall i \in [\![n]\!], \hat{u}_i^{(t)} = u_i\big(\pi^{(t)}\big) = \frac{1}{c_i^{(t)}} \sum_{\tau \le t} \mathbb{1}_{\{i^{(\tau)}=i\}} \langle \mu_i | e^{(\tau)} \rangle, \\ &\forall j \in [\![m]\!], \hat{v}_j^{(t)} = \sum_{i=1}^n \hat{w}_i^{(t)} \pi_{ij}^{(t)} = \frac{1}{t} \sum_{\tau=1}^t e_j^{(\tau)}. \end{aligned} \tag{16}$$

(Recall $e^{(t)} = E(\sigma^{(t)})$.) This gives the formula computed in Alg. 1 for $\hat{g}_i^{(t)} = g_{\hat{w}^{(t-1)},i}\big(\pi^{(t-1)}\big)$.

For the online updates of $\hat{u}$ and $\hat{v}$, we use as initial value $\hat{u}_i^{(0)}$ the utility of the random ranking and $\hat{v}_j^{(0)} = 0$. Since $\hat{u}_i^{(t)}$ only changes for $i^{(t)} = i$, they are given by:

$$\begin{aligned} &\forall i, \hat{u}_i^{(0)} = \langle \mu_i | \frac{\|b\|_1}{m} \rangle \\ &\hat{u}_{i^{(t)}}^{(t)} = \hat{u}_{i^{(t)}}^{(t-1)} + \frac{1}{t}\Big(\langle \mu_i | e^{(t)} \rangle - \hat{u}_{i^{(t)}}^{(t-1)}\Big) \\ &\hat{v}^{(t)} = \hat{v}^{(t-1)} + \frac{1}{t}\Big(e^{(t)} - \hat{v}^{(t-1)}\Big). \end{aligned} \tag{17}$$

Quality-weighted exposure. Similarly, for quality-weighted exposure (7), approximate gradients $g_{\hat{w}^{(t)},i}(\pi^{(t)})$ use online estimates of exposures $\hat{v}^{(t)}$ as in (17), as well as online estimates of the qualities using $\forall j, \hat{q}_j^{(0)} = 0$:

$$\begin{aligned} &\hat{q}^{(t)} = \frac{1}{t}\sum_{\tau=1}^t \mu_{i^{(t)}} = \hat{q}^{(t-1)} + \frac{1}{t}\Big(\mu_{i^{(t)}} - \hat{q}^{(t-1)}\Big), \\ &\hat{q}_{\text{avg}}^{(t)} = \frac{1}{m}\sum_{j=1}^m \hat{q}_j^{(t)}. \end{aligned} \tag{18}$$

Balanced exposure. Balanced exposure to user groups (8) works similarly, except that we need to keep track of user counts within each group, which we denote by $c_s^{(t)}$, as well as exposures within each group:

$$\begin{aligned} &\forall j \in [\![m]\!], c_s^{(t)} = \sum_{i \in s} c_i^{(t)}, \\ &\hat{v}_{j|s}^{(t)} = \frac{1}{c_s^{(t)}} \sum_{\tau=1}^t \mathbb{1}_{\{i^{(\tau)} \in s\}} e_j^{(\tau)}, \\ &\hat{v}_{j|\text{avg}}^{(t)} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \hat{v}_{j|s}^{(t)}. \end{aligned} \tag{19}$$

We use $\hat{v}_{j|s}^{(t)} = 0$ if $c_s^{(t)} = 0$ since the item has not been exposed to a group we never saw. As for $\hat{v}$ and $\hat{q}$, these counts are updated online in $O(m)$ operations because they only change for the group of user $i^{(t)}$.

The guarantees we obtain for these algorithms are the following. The proof is given in App. B.

Proposition 3. *The approximate gradients of Alg. 1, 2 and 3 satisfy:*

$$\textit{Alg. 1: } \mathbb{E}\Big[\Big\|\hat{g}_i^{(t)} - g_{w,i}\big(\pi^{(t-1)}\big)\Big\|_\infty\Big] \le \frac{\beta\|\psi''_{\alpha_2}\|_\infty}{m}\sqrt{\frac{n}{t-1}},$$

$$\textit{Alg. 2 } \mathbb{E}\Big[\Big\|\hat{g}_i^{(t)} - g_{w,i}\big(\pi^{(t-1)}\big)\Big\|_\infty\Big] \le \frac{\beta\big(2+\|b\|_1\big)^2}{m\min(\eta,\sqrt{\eta})}\sqrt{\frac{n}{t-1}},$$

Input at time t: user index i
// Step (1)
Compute score for each item j:
$\hat{g}_{ij} = \psi'_{\alpha_1}(\hat{u}_i)\mu_{ij} \quad \frac{\beta}{m}\psi'_{\alpha_2}(\hat{v}_j)$
// Step (2)
Update $\hat{u}_i$ and all $\hat{v}_j$ using (17)
// Step (3)
Return $\sigma = \text{topk}(\hat{g}_i)$

Alg. 1: OFFR/two-sided fairness (6).

Input at time t: user index i
// Step 1
Compute score for each item j:
$\hat{g}_{ij} = \mu_{ij} - \frac{\beta}{m\hat{Z}}\Big(\hat{q}_{\text{avg}}\hat{v}_j - \hat{q}_j\|b\|_1\Big)$
where $\hat{Z} = \sqrt{\eta \quad \frac{1}{m}\sum_{j=1}^{m}\Big(\hat{q}_{\text{avg}}\hat{v}_j - \hat{q}_j\|b\|_1\Big)^2}$
// Step (2)
Update all $\hat{v}_j, \hat{q}_j, \hat{q}_{\text{avg}}$ using (18)
// Step (3)
Return $\sigma = \text{topk}(\hat{g}_i)$

Alg. 2: OFFR/quality-weighted (7).

Input at time t: user index i
// Step 1 ($s[i]$ is the group of user i)
Compute score for each item j:
$\hat{g}_{ij} = \mu_{ij} - \frac{\beta}{m\hat{Z}_j}.\frac{t}{c_{s[i]}} \, 1 \Big(\hat{v}_{j|s[i]} - \hat{v}_{j|\text{avg}}\Big)$
where $\hat{Z}_j = \sqrt{\eta \quad \sum_{s\in\mathcal{S}}\Big(\hat{v}_{j|s} - \hat{v}_{j|\text{avg}}\Big)^2}$
// Step (2)
Update $\forall j, \hat{v}_{j|s[i]}$ using (19)
// Step (3)
Return $\sigma = \text{topk}(\hat{g}_i)$

Alg. 3: OFFR/balanced exposure (8).

Alg. 3 $\mathbb{E}\Big[\Big\|\hat{g}_i^{(t)} - g_{w,i}\big(\pi^{(t-1)}\big)\Big\|_\infty\Big] \leq \dfrac{\beta\Big(\frac{1}{\overline{w}_{s[i]}t} + 8\sum_{s\in\mathcal{S}}\sqrt{\frac{|s|}{\overline{w}_s(t-1)}}\Big)}{m\overline{w}_{s[i]}\min(\eta, \sqrt{\eta})}.$

Overall, they all decrease in $O(\frac{1}{\sqrt{t}})$ as desired to apply our convergence result Th. 2. More interestingly, the bounds do not depend on w, which means that the objectives are well-behaved even when some users have low probabilities. The balanced exposure criterion does depend on $\frac{1}{\overline{w}_s}$, which means that the bound becomes arbitrarily bad when some groups have small cumulative activity. This is natural, since achieving balanced exposure across groups dynamically is necessarily a difficult task if one group is only very rarely observed.

Putting together Thm. 2 and Prop. 3, we obtain regret bounds of order $1/\sqrt{t}$:

COROLLARY 4. *Ignoring constants and assuming $\eta \leq 1$, the regrets $R(t)$ of Alg. 1, 2 and 3 are bounded as in Table 1.*

Algorithm	Order of magnitude of $R^{(t)}$
Alg. 1	$\Big(\sqrt{n}\|b\|_1 + \frac{n\|b\|_1\beta}{m}\Big)\Big(\eta^{\alpha_1-1} + \eta^{\alpha_2-2}\Big)\sqrt{\frac{1}{t}}$
Alg. 2	$\Big(\sqrt{n}\|b\|_1 + \frac{n\|b\|_1^3\beta}{m}\Big)\eta^{-1}\sqrt{\frac{1}{t}}$
Alg. 3	$\Big(\sqrt{n}\|b\|_1 + \frac{n\|b\|_1\beta}{m}\sqrt{\frac{\lvert\mathcal{S}\rvert}{\overline{w}_{\min}^3}}\Big)\eta^{-1}\sqrt{\frac{1}{t}}$

Table 1: Upper bounds on regret $R(t)$, ignoring constants and assuming $\eta \leq 1$. In all cases, we have $R(t) = O(1/\sqrt{t})$. For balanced exposure, the regret bound also depends on the minimum total weight of a group $\overline{w}_{\min} = \min_{s\in\mathcal{S}}\overline{w}_s$.

The proof of Cor. 4 is given in App. C. Compared to a batch Frank-Wolfe algorithm for the same objectives, we obtain a convergence in $O(1/\sqrt{t})$ instead of $1/t$ [11, Prop 4.]. Part of this difference is due to the variance in the gradients due to unknown user activities, but Th. 2 would be of order $1/\sqrt{t}$ even with true gradients (i.e., $D_i = 0$). We do not believe our bound can be improved because our online setting we consider is only equivalent to a Frank-Wolfe algorithm if we consider a stochastic "stepsize" of $1/c_i^{(t)}$ for user i at time step t (which yields the average exposures $\pi^{(t)}$, our object of study), which introduces additional variance in the optimization. We leave the proof of lower bounds to future work.

4.2 Computational complexity

To simplify the discussion on computational complexity, we assume the number of groups in balanced exposure is $O(1)$ (i.e., negligible compared to n and m), which is the case in practice for groups such as gender or age. For each of the algorithms, the necessary computations involve $O(m)$ floating point operations to compute the scores of all items (step (1)), $O(m + k\ln k)$ (amortized) comparisons for the top-k sort (step (3)). The update of the online estimates (step (2)) requires $O(m)$ operations. More involved implementations of this step require only $O(k)$ operations by only updating the recommended items, but they require additional computations in step (1), which remains in $O(m)$. In all cases, the computation cost is dominated by the top-k sort, which would likely required in practice even without consideration for fairness of exposure. Thus, OFFR provides a general approach to fairness of exposure in online ranking that does not involve significantly more computations than having no considerations for fairness of exposure at all, despite optimizing an objective function where the optimal ranking of each user depends on the rankings of all other users.

Memory requirements. For two-sided fairness (Alg. 1), we need $O(n + m)$ bytes for $\hat{u}$ and $\hat{v}$. For quality weighted exposure we need $O(m)$ bytes to store $\hat{v}$ and $\hat{q}$, while we need $O(m|\mathcal{S}|)$ bytes for balanced exposure. In all cases, storage is of the order $O(n + m)$. Notice that in practice, it is likely that counters of item exposures and user utility are computed to monitor the performance of the system anyway. The additional storage of our algorithm is then negligible.

5 EXPERIMENTS

We provide in this section experiments on simulated ranking tasks following the protocol of Do et al. [11]. Our experiments have two goals. First we study the convergence of OFFR to the desired trade-off values for the three objectives of Sec. 2.2, by comparing objective function values of Offr and the batch Frank-Wolfe algorithm for fair ranking of [11] at comparable computational budgets. Second, we compare the dynamics of Offr and *FairCo* [32], an online ranking algorithm designed to asymptotically achieve equal quality-weighted exposure for all items. We also provide a comparison between Offr and *FairCo* on balanced exposure by proposing an ad-hoc extension to *FairCo* for that task. In the three next subsections, we first describe our experimental protocol (Subsection 5.1). Then, we give qualitative results in terms of the trade-offs achieved by Offr by varying the weight of the exposure objective (Subsection 5.2). We finally we dive into the comparison between Offr and batch Frank-Wolfe (Subsection 5.3) and between Offr and *FairCo* (Subsection 5.4).

5.1 Experimental setup

Data. We use the Last.fm dataset of Celma [8], which includes $360k$ users and $180k$ items (artists), from which we select the top $15k$ users and $15k$ items having the most interactions. We refer to this subset of the dataset as *lastfm15k*. The (user, item) values are estimated using a standard matrix factorization for learning from positive feedback only Hu et al. [17]. Details of this training can be found in App. D.1. Since we focus on ranking given the preferences rather than on the properties of the matrix factorization algorithm, we consider these preferences as our ground truth and given to the algorithm, following previous work[9, 11, 33, 40]. In App. D.3, we present results on the MovieLens dataset [14]. The results are qualitatively similar. Both datasets come with a "gender" table associated to user IDs. It is a ternary value 'male', 'female', 'other' (see [8, 14] for details on the datasets). On *lastfm15k*, the resulting dataset contains $10k$/ $3.6k$/$1.4k$ users of category 'male'/'female'/'other' respectively.

Tasks. We study the three tasks described in Sec. 2.2: two-sided fairness, quality-weighted exposure and balanced exposure to user groups. Note that it is possible to study weighted combinations of these objective, since the combined objective would remain concave and smooth. We focus on the three canonical examples to keep the exposition simple. We use the gender category described above as user groups, and they are only used for the last objective. We study the behavior of the algorithm as we vary $\beta > 0$, which controls the trade-off between user utility and item fairness. For two-sided fairness, we take $\alpha_1 = \alpha_2 = 0$ in (6), which are recommended values in Do et al. [11] to generate trade-offs between user and item fairness. In all cases, we use $\eta = 1$ in this section, and show the results for $\eta = 0.01$ (a less smooth objective function) in App. D.2. We assume that the true user activities are uniform (but the online algorithm does not know about these activities). We consider top-k rankings with $k = 40$. We set the exposure weights to $b_r = \frac{1}{\log_2(1+r)}$, which correspond to the well-known DCG measure, as in [3, 11, 33]. In the following, we use the term *iteration* to refer to a time step, and *epoch* to refer to n timesteps (which correspond to the order of magnitude of time steps required to see every user). Notice that in the online setting, users are sampled with replacement at each iteration following our formal framework of Sec. 2, so the online algorithms are not guaranteed to see every user at every epoch.

Comparisons. We compare to two previous works:

(1) Batch Frank-Wolfe (*batch-FW*): The only tractable algorithm we know of for all these objectives is the *static* algorithm of Do et al. [11]. We compare offline vs online learning in terms of convergence to the objective value for a given computational budget. The algorithm of Do et al. [11] is based on Frank-Wolfe as well, which we refer to as *batch-FW* and our approach (referred to as Offr) is an online version of *batch-FW*. Thus the cost per user per epoch (one top-k sort) are the same. We use this baseline to benchmark how fast Offr convergence to the optimal value.
(2) *FairCo* [32]: We use the approach from [32] introduced for quality-weighted exposure in dynamic ranking. In our notation, dropping the time superscrits, given user i at time step t, *FairCo* outputs

$$(\textit{FairCo}\ [32]) \qquad \sigma = \mathrm{topk}(\tilde{\mu}) \qquad \text{with } \tilde{\mu}_j = \mu_{ij} \quad \beta(t-1)\max_{j'}\left(\frac{\hat{v}_{j'}}{\hat{q}_{j'}} - \frac{\hat{v}_j}{\hat{q}_j}\right) \tag{20}$$

where β trades-off the importance of the user values μ_{ij} and the discrepancy between items in terms of quality-weighted exposure. Notice that the item realizing the maximum in (20) is the same for all j, so the computational complexity of *FairCo* is similar to that of Offr.
A fundamental difference between *FairCo* is that the weight given to the fairness objective increases with t. The authors proved in the paper that the average $\frac{1}{m(m-1)}\sum_{j,j'}\left|\frac{\hat{v}_{j'}}{\hat{q}_{j'}} - \frac{\hat{v}_j}{\hat{q}_j}\right|$ converges to 0 at a rate $O(1/t)$. However, they do not discuss the convergence of the user utilities depending on β. Fundamentally, *FairCo* and Offr address different problems, since Offr aims for trade-offs where the relative weight of the user objective and the item objective is fixed from the start. Even though they should converge to different outcomes, we compare the intermediate dynamics at the early stage of optimization.

In addition, we compare to an extension of *FairCo* to balanced exposure. Even though *FairCo* was not designed for balanced exposure, we propose to follow a similar recipe as (20) as baseline for balanced exposure:

$$\sigma = \mathrm{topk}(\tilde{\mu}) \quad \text{with } \tilde{\mu}_j = \mu_{ij} \quad \beta(t-1)\max_{s\in\mathcal{S}}\left(\hat{v}_{j|s} - \hat{v}_{j|s[i]}\right) \tag{21}$$

All our experiments are repeated and averaged on three seeds for sampling the users at each step. The online algorithms are run for 5000 epochs, and the batch algorithms for $50,000$ epochs.

5.2 Qualitative results: effect of varying β

We first present qualitative results regarding the trade-offs that are obtained by varying the weight of the fairness penalty β from 0.001 to 100 by powers of 10, for all three tasks in Fig. 1. The y-axis is the user objective for two-sided fairness and the average user utility for quality-weighted and balanced exposure. The x-axis is the item objective (higher is better) for two-sided fairness, and the item penalty term with $\eta = 0$ of (7) and (8) for quality-weighted and balanced exposure respectively.

At a high level, we observe as expected a Pareto front spanning a large range of (user, item) objective values. We also observe on all

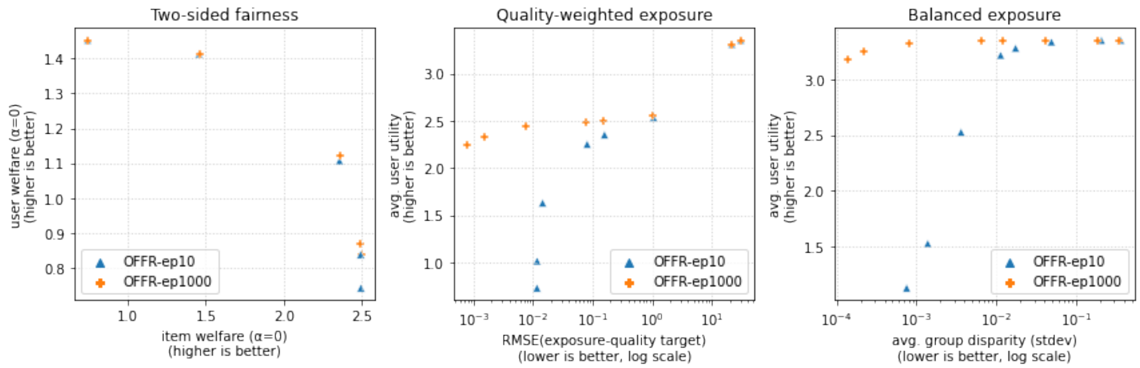

Figure 1: Trade-offs between user objective (y-axis) and item fairness (x-axis), at the beginning of the online process (10 epochs) and closer to the end (1000 epochs). For two-sided fairness, both user and item objectives should be maximized, while for quality-weighted balanced exposure the item objectives (x-axis) should be minimized. As expected, varying the weight of the item objective β leads to different trade-offs between user utility and item exposure. Comparing epochs 10 and 1000 on quality-weighted and balanced exposure, we observe that with large β, Offr tends to prioritize the item objective and has low user utility at the beginning of training.

three tasks but specifically quality-weighted and balanced exposure that at the beginning of training (epoch 10), the item objective values are close to the final values, but the user objective values increase a lot on the course of training. We will get back to this observation in our comparison with *FairCo*. As anecdotal remarks, we first observe that on two-sided ranking, convergence is very fast and the trade-offs obtained at epoch 10 and 1000 are relatively close. Second, we observe that for balanced exposure on this dataset, it is possible to achieve near perfect fairness (item objective $\leq 10^{-3}$) at very little cost of user utility at the end of training.

5.3 Online convergence

To compare the convergence of Offr compared to *batch-FW*, Fig. 2 plots the regret, i.e., the difference between the maximum value and the obtained objective value on the course of the optimization for both algorithms,[4] in log-scale as a function of the number of epochs for the three tasks for $\beta \in \{0.01, 1.0\}$. We first observe that convergence is slower for larger values of β, which is coherent with the theoretical analysis. We also observe that for the first 1000 epochs (recall that an epoch has the same compatational cost for both algorithms), Offr fares better than the batch algorithm. Looking at more epochs or different values of η (shown in Fig. 4 and 6 in App. D.2), we observe that *batch-FW* eventually catches up. This is coherent with the theoretical analysis, as the *batch-FW* converges in $1/t$ [11], but Offr in $O(1/\sqrt{t})$. In accordance with the well-known performance of stochastic gradient descent in machine learning [4], the online algorithm seems to perform much better at the beginning, which suggests that it is practical to run the algorithm online.

[4]The maximum value for the regret is taken as the maximum between the result of *batch-FW* after 50k epochs and Offr after 5k epochs. For both Offr and *batch-FW*, we compute the ideal objective function, i.e., knowing the user activities w. Notice that Offr does not know w but *batch-FW* does.

5.4 Comparison to *FairCo*

We give in Fig. 3 illustrations of the dynamics of Offr compared to *FairCo* [32] described in (20) and (21). Since *FairCo* aims at driving a disparity to 0, it cannot be applied to two-sided fairness, so we focus on quality-weighted and balanced exposure. Contrarily to the previous section, we cannot compare objective functions at convergence or convergence rates because *FairCo* does not optimize an objective function. Our plots show the item objective (x-axis, lower is better, log-scale) and the average user utility on the y-axis. Then, for Offr and *FairCo* for two values of β, we show the (item objective, user utility) values obtained on the course of the algorithm. For *FairCo*, we chose $\beta = 0.001$ which gives overall the highest user utility values we observed, and $\beta = 1$, a representative large value. For Offr we chose two different values of β that achieve different trade-offs. This choice has no impact on the discussion. The left plots show vanilla Offr, the right plots show Offr with a "pacing" heuristic described later.

Convergence properties. Looking at the left plot, we see Offr converging to its trade-off dictated by the value of β. On the other hand, *FairCo* does not converge. As expected, as time goes by, *FairCo* reduces the item objective to low values. Interestingly though, it seems that the average user utility *seems to* converge for *FairCo* to a value that depends on β. It is likely an effect of the experimental setup: with $\beta = 0.001$, *FairCo* is far from the regime where it achieves low values of the item objective within our 5000 epochs (as seen by the discrepancy in item objective between $\beta = 1$ and $\beta = 0.001$). Overall, since *FairCo* does not have a clear objective function nor theoretical guarantees regarding the user utility, *FairCo* does not allow to choose the trade-off between user and item objectives that is desired. On the bright side, *FairCo* does happen to reduce the item objective to very low values for $\beta = 1$ as the number of iteration decreases.

Trade-offs. Interestingly, *FairCo* and Offr have different dynamics. The plots show that on the course of the iterations, Offr rapidly

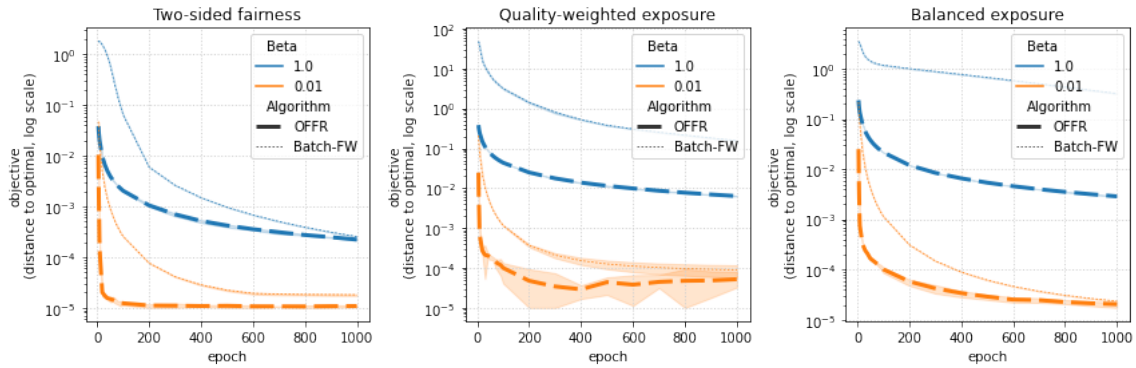

Figure 2: Convergence speed of Offr compared to *batch-FW* on the three fairness objectives, for $\beta \in \{0.01, 1\}$ and $\eta = 1$. The y-axis is the regret in log-scale. We observe that Offr is faster than *batch-FW* at the beginning, especially for large β.

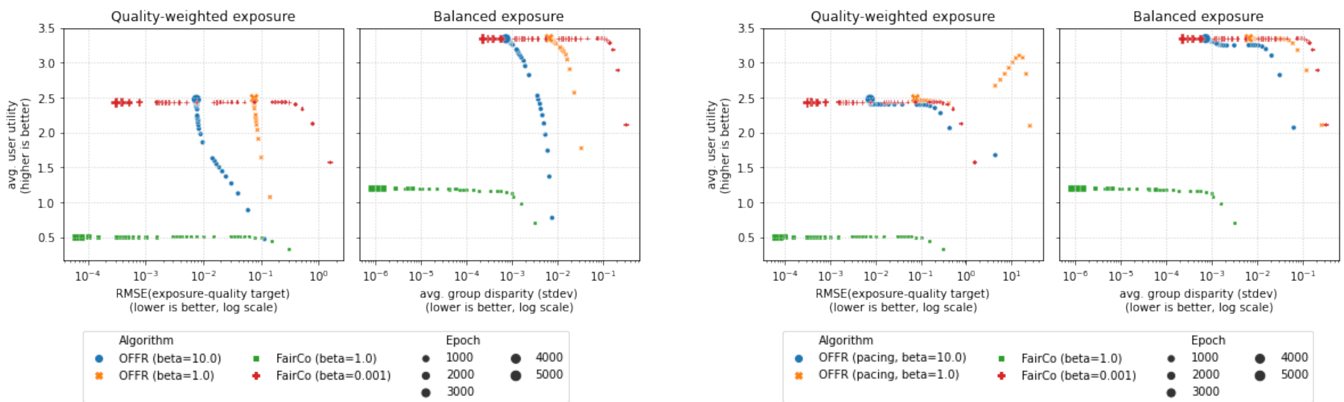

Figure 3: Convergence of Offr compared to the dynamics of *FairCo*. Each point is the average user utility (y-axis) vs item objective (x-axis, log-scale) for an algorithm and value of β (color/marker), at a given epoch (the size of the markers increase with the epoch number). The trajectory describes the online dynamics of each algorithm in terms of the trade-offs they achieve. (left) Offr converges to the trade-off dictated by its value of β while keeping its item objective near the target value from the beginning, increasing the user utility with time. (right) the pacing heuristic added to Offr provides a way to approach the final trade-off while keeping user utility high during the entire course of optimization.

reaches its item objective, but takes time to reach its user objective (as seen by the "vertical" pattern of Offr in the left plot of Fig. 3, which means that the item objective does not change a lot). In contrast, *FairCo* for small β starts from high user utility and decreases the item objective from there. Evidently, Offr and *FairCo* strike different trade-offs, and neither of them is universally better: it depends on whether we prioritize user utility or item fairness at the early stages of the algorithm. Nonetheless, to emulate *FairCo*'s trade-offs, we propose a "pacing" heuristic which uses a time-dependent β in our objective, using $\beta_t = \min(\beta, \gamma . \frac{t}{n})$ where $\gamma > 0$ is the pacing factor. the right plots of Fig. 3 show the results with $\gamma = 0.01$. We observe now a more "horizontal" pattern in the dynamics of Offr, similarly to *FairCo*, meaning that Offr sucessfully pioitizes user utility over item fairness in the early stages of the algorithm. Whether or not such a pacing should be used depends on the requirements of the application.

6 RELATED WORK

The question of the social impact of recommender systems started with independent audits of bias against groups defined by sensitive attributes [13, 21, 26, 29, 37]. Algorithms for fairness of exposure have been studied since then [3, 6, 7, 11, 32, 34, 42]. The goal is often to prevent winner-take-all effects or popularity bias [1, 34] or promote smaller producers to incentivize production [27, 30, 31].

The question of online ranking is often studied in conjunction with learning to rank, i.e., learning the (user, item) values μ. The previous work by Morik et al. [32], which we compare to in the experiments (the *FairCo* baseline), had this dimension, which we do not. On the other hand, as we discussed, their algorithm has limited scope because it only aims at asymptotically removing any disparity. The algorithm cannot be used on other forms of loss function such as two-sided fairness, and cannot be used to converge to intermediate trade-offs. Their theoretical guarantee is also relatively weak, since they only prove that the exposure objective converges to 0, without any guarantee on the user utility. In contrast, we show that the regret of our algorithm converges to 0 for a wide range of objectives.

Yang and Ai [41] also proposes an online algorithm combining learning to rank and fairness of exposure, but they compare the exposure of groups of items within single rankings, as opposed to considering exposure of items across users. Their fairness criterion does not involve the challenge we address, since the optimal rankings in their case can still be computed individually for every user.

Recently, fairness of exposure has been studied in the bandit setting [19, 28]. These works provide experimental evaluations of bandit algorithms with fairness constraints, but they do not provide theoretical guarantees. Wang et al. [39] also consider fairness of exposure in bandits, but without ranking.

Compared to this literature on dynamic ranking, we decided to disentangle the problem of learning the user preferences from the problem of generating the rankings online while optimizing a global exposure objective. We obtain a solution to the ranking problem that is more general than what was proposed before, with stronger theoretical guarantees. Our approach unlocks the problem of online ranking with a global objective function, and we believe that our approach is a strong basis for future exploration/exploitation algorithms.

We studied online ranking in a stationary environment. Several works consider multi-step recommendations scenarios with dynamic models of content production [31, 43]. They study the effect of including an exposure objective on the long-term user utilities, but they do not focus on how to efficiently generate rankings.

Relationship to Frank-Wolfe algorithms. The problem of inferring ranking lies in between convex bandit optimization [see 2, and references therein] and stochastic optimization. Our problem is easier than bandit optimization since the function is known – at least partially, and in all cases there is no need for active exploration. The main ingredient we add to the convex bandit optimization literature is the multi-user structure, where parameters are decomposed into several blocks that can only be updated one at a time, while optimizing for a non-decomposable objective function. The similarity with the bandit optimization algorithm of Berthet and Perchet [2] is the usage of the Frank-Wolfe algorithm to generate a deterministic decision at each step while implicitly optimizing in the space of probability distributions.

Our algorithm is a Frank-Wolfe algorithm with a stochastic gradient [15, 25] and block-separable constraints [22, 24]. The difference with this line of work is twofold. First, the distribution w is not necessarily uniform. Second, in our case, different users have different "stepsizes" for their parameters (the stepsize is $\frac{1}{c_i^{(t)}}$ for the user i sampled at time t), rather than a single predefined stepsize. These two aspects complicate the analysis compared to that of the stochastic Frank-Wolfe with block-separable constraints of Lacoste-Julien et al. [24].

7 CONCLUSION AND DISCUSSION

We presented a general approach to online ranking by optimizing trade-offs between user performance and fairness of exposure. The approach only assumes the objective function is concave and smooth. We provided three example tasks involving fairness of exposure, and the scope of the algorithm is more general. For instance, it also applies to the formulation of Do et al. [11] for reciprocal recommendation tasks such as dating applications.

Despite the generality of the framework, there are a few technical limitations that could be addressed in future work. First, the assumption of the position-based model (1) is important in the current algorithmic approach, because it yields the linear structure with respect to exposure that is required in our Frank-Wolfe approach. Dealing with more general cascade models [10, 28] is an interesting open problem. Second, we focused on the problem of generating rankings, assuming that (user, item) values μ_{ij} are given by an oracle and are stationary over time. Relatedly to this stationarity assumption, we ignored the feedback loops involved in recommendation. These include feedback loops due to learning from prior recommendations [5], the impact of the recommender system on users' preferences themselves [20], as well as the impact that fairness interventions on content production [31]. Third, our approach to balanced exposure is based on the knowledge of a discrete sensitive attribute of users. Consequently, this criterion cannot be applied when there are constraints on the direct usage of the sensitive attribute within the recommender system, when the sensitive attribute is not available, or when the delineation of groups into discrete categories is not practical or ethical [38].

Finally, while we believe fairness of exposure is an important aspect of fairness in recommender systems, it is by no means the only one. For instance, the alignment of the system's objective with human values [36] critically depends on the definition of the quality of a recommendation, the values μ_{ij} in our framework. The fairness of the underlying system relies on careful definitions of these μ_{ij}s and on unbiased estimations of them from user interactions – in particular, taking into account the non-stationarities and feedback loops mentioned previously.

ACKNOWLEDGMENTS

The authors thank Alessandro Lazaric and David Lopez-Paz for their feedback on the paper.

REFERENCES

[1] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The Unfairness of Popularity Bias in Recommendation. In *RecSys Workshop on Recommendation in Multistakeholder Environments (RMSE)*.
[2] Quentin Berthet and Vianney Perchet. 2017. Fast Rates for Bandit Optimization with Upper-Confidence Frank-Wolfe. In *Advances in Neural Information Processing Systems*.
[3] Asia J Biega, Krishna P Gummadi, and Gerhard Weikum. 2018. Equity of attention: Amortizing individual fairness in rankings. In *ACM SIGIR conference on research & development in information retrieval*.
[4] Léon Bottou and Olivier Bousquet. 2008. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems*.
[5] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. *Journal of Machine Learning Research* 14, 65 (2013), 3207–3260.
[6] Robin Burke. 2017. Multisided Fairness for Recommendation. arXiv:1707.00093 [cs.CY]
[7] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. 2018. Ranking with Fairness Constraints. arXiv:1704.06840 [cs.DS]
[8] O. Celma. 2010. *Music Recommendation and Discovery in the Long Tail.* Springer.

[9] Abhijnan Chakraborty, Gourab K Patro, Niloy Ganguly, Krishna P Gummadi, and Patrick Loiseau. 2019. Equality of voice: Towards fair representation in crowd-sourced top-k recommendations. In *ACM Conference on Fairness, Accountability, and Transparency*.
[10] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An experimental comparison of click position-bias models. In *International conference on web search and data mining*.
[11] Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. 2021. Two-sided fairness in rankings via Lorenz dominance. In *Conference on Neural Information Processing Systems*.
[12] Marguerite Frank and Philip Wolfe. 1956. An algorithm for quadratic programming. *Naval research logistics quarterly* 3, 1-2 (1956), 95–110.
[13] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. 2014. Measuring price discrimination and steering on e-commerce web sites. In *ACM Internet Measurement Conference*.
[14] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2015), 1–19.
[15] Elad Hazan and Satyen Kale. 2012. Projection-free online learning. In *International Coference on International Conference on Machine Learning*.
[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *International Conference on World Wide Web*.
[17] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining*.
[18] Basileal Imana, Aleksandra Korolova, and John Heidemann. 2021. Auditing for Discrimination in Algorithms Delivering Job Ads. In *The Web Conference*.
[19] Olivier Jeunen and Bart Goethals. 2021. Top-k contextual bandits with equity of exposure. In *ACM Conference on Recommender Systems*.
[20] Dimitris Kalimeris, Smriti Bhagat, Shankar Kalyanaraman, and Udi Weinsberg. 2021. Preference Amplification in Recommender Systems. In *ACM Conference on Knowledge Discovery & Data Mining*.
[21] Matthew Kay, Cynthia Matuszek, and Sean A Munson. 2015. Unequal representation and gender stereotypes in image search results for occupations. In *ACM Conference on Human Factors in Computing Systems*.
[22] Thomas Kerdreux, Fabian Pedregosa, and Alexandre d'Aspremont. 2018. Frank-Wolfe with subsampling oracle. In *International Conference on Machine Learning*.
[23] Yehuda Koren and Robert Bell. 2015. Advances in collaborative filtering. *Recommender systems handbook* (2015), 77–118.
[24] Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. 2013. Block-coordinate Frank-Wolfe optimization for structural SVMs. In *International Conference on Machine Learning*.
[25] Jean Lafond, Hoi-To Wai, and Eric Moulines. 2016. On the Online Frank-Wolfe Algorithms for Convex and Non-convex Optimizations. arXiv:1510.01171 [stat.ML]
[26] Anja Lambrecht and Catherine Tucker. 2019. Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads. *Management Science* 65, 7 (2019), 2966–2981.
[27] Weiwen Liu, Jun Guo, Nasim Sonboli, Robin Burke, and Shengyu Zhang. 2019. Personalized fairness-aware re-ranking for microlending. In *ACM Conference on Recommender Systems*.
[28] Masoud Mansoury, Himan Abdollahpouri, Bamshad Mobasher, Mykola Pechenizkiy, Robin Burke, and Milad Sabouri. 2021. Unbiased Cascade Bandits: Mitigating Exposure Bias in Online Learning to Rank Recommendation. arXiv:2108.03440 [cs.IR]
[29] Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. 2017. Auditing search engines for differential satisfaction across demographics. In *International Conference on World Wide Web Companion*.
[30] 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 *ACM International Conference on Information and Knowledge Management*.
[31] Martin Mladenov, Elliot Creager, Omer Ben-Porat, Kevin Swersky, Richard Zemel, and Craig Boutilier. 2020. Optimizing long-term social welfare in recommender systems: A constrained matching approach. In *International Conference on Machine Learning*.
[32] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling fairness and bias in dynamic learning-to-rank. In *ACM SIGIR Conference on Research and Development in Information Retrieval*.
[33] Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P Gummadi, and Abhijnan Chakraborty. 2020. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *The Web Conference*.
[34] 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.
[35] Ashudeep Singh and Thorsten Joachims. 2019. Policy Learning for Fairness in Ranking. In *Conference on Neural Information Processing Systems*.
[36] Jonathan Stray, Ivan Vendrov, Jeremy Nixon, Steven Adler, and Dylan Hadfield-Menell. 2021. What are you optimizing for? Aligning Recommender Systems with Human Values. arXiv:2107.10939 [cs.IR]
[37] Latanya Sweeney. 2013. Discrimination in online ad delivery. *Queue* 11, 3 (2013), 10.
[38] Nenad Tomasev, Kevin R. McKee, Jackie Kay, and Shakir Mohamed. 2021. Fairness for Unobserved Characteristics: Insights from Technological Impacts on Queer Communities. In *AAAI/ACM Conference on AI, Ethics, and Society*.
[39] Lequn Wang, Yiwei Bai, Wen Sun, and Thorsten Joachims. 2021. Fairness of Exposure in Stochastic Bandits. arXiv:2103.02735 [cs.LG]
[40] Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. 2021. TFROM: A Two-Sided Fairness-Aware Recommendation Model for Both Customers and Providers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*.
[41] Tao Yang and Qingyao Ai. 2021. Maximizing Marginal Fairness for Dynamic Learning to Rank. In *The Web Conference*.
[42] Meike Zehlike and Carlos Castillo. 2020. Reducing disparate exposure in ranking: A learning to rank approach. In *The Web Conference*.
[43] Ruohan Zhan, Konstantina Christakopoulou, Ya Le, Jayden Ooi, Martin Mladenov, Alex Beutel, Craig Boutilier, Ed Chi, and Minmin Chen. 2021. Towards Content Provider Aware Recommender Systems: A Simulation Study on the Interplay between User and Provider Utilities. In *The Web Conference*.