FAIRNESS OF EXPOSURE FOR RANKING SYSTEMS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Ashudeep Singh

August 2021

FAIRNESS OF EXPOSURE FOR RANKING SYSTEMS

Ashudeep Singh, Ph.D.

Cornell University 2021

Ranking-based interfaces are ubiquitous in today's multi-sided online economies (such as online marketplaces, job search, property renting, media streaming). In these systems, the items to be ranked are products, job candidates, creative content, or other entities that transfer economic benefit. It is widely recognized that the position of an item in the ranking has a crucial influence on its exposure which directly translates into economic opportunity. Surprisingly, learning-to-rank (LTR) approaches typically do not consider their impact on the opportunity they provide to the items. Instead, most LTR algorithms solely focus on maximizing the utility of the rankings to the user issuing the query, while there is evidence that this does not necessarily lead to rankings that would be considered fair or desirable in many situations.

This thesis proposes a conceptual and computational framework that allows the formulation of fairness constraints on rankings in terms of a merit-based exposure allocation. As a part of this framework, we develop efficient learning-to-rank algorithms that maximize the utility for the user while provably satisfying a specifiable notion of fairness. Since fairness goals can be application-specific, we show that a broad range of fairness constraints can be implemented in this framework using its expressive power to link relevance, merit, exposure, and impact. Beyond the theoretical evidence in deriving the frameworks and algorithms, empirical results on simulated and real-world datasets verify the effectiveness of the approach on both individual and group-fairness notions.

BIOGRAPHICAL SKETCH

Ashudeep Singh was born in Jalandhar, a city in Punjab, India, in the year 1993. He grew up in the town of Mohali, Punjab, and went to school in the city of Chandigarh before moving to Kanpur for his undergraduate studies. He holds a B.Tech. & M.Tech. in Computer Science and Engineering from Indian Institute of Technology (IIT) Kanpur. He is currently a Ph.D. candidate in the Department of Computer Science at Cornell University, advised by Prof. Thorsten Joachims.

Ashudeep is broadly interested in designing machine learning algorithms for information retrieval and recommender systems that learn from interactive user feedback. He loves to think about the impact of online platforms on users as well as creators, producers, and sellers, and about building algorithms that lead to a fairer distribution of economic opportunity in such systems.

During his time at Cornell, he completed internships at Microsoft Research in New York City and Montréal, Facebook in Menlo Park, and Google in New York City. Upon receiving his Ph.D., he will join the Applied Science group at Pinterest in Palo Alto, CA, to develop machine learning algorithms for recommendations that are fair and equitable to creators and users.

Outside of work, Ashudeep enjoys playing and watching soccer. He loves and cherishes nature and enjoys camping, hiking, and photography.

Dedicated to my grandmother.

ACKNOWLEDGEMENTS

First and foremost, I am deeply indebted to my Ph.D. advisor Thorsten Joachims whose constant support, guidance, and thoughtful advice is the reason I am able to make this contribution to the field. Throughout graduate school, he was kind and patient, and I will forever be grateful for his phenomenal mentorship. I am also very thankful to my other committee members – David Mimno, Karthik Sridharan, and Solon Barocas. Their insightful feedback and questions have brought both novelty and clarity to this work. Solon Barocas' class "Ethics and Policy in Data Science" and his work introducing the concepts of Fairness, Accountability, and Transparency to the area of machine learning were very stimulating and inspiring as I started to do research in this direction.

During my Ph.D., I was fortunate to taste research in the industry through research internships working with incredible mentors. I am grateful to Alex Beutel, Fernando Diaz, Khalid El-Arini, and John Langford for being exceptional internship hosts, supporting my research ideas, and giving me valuable career advice. Your work towards solving challenging scientific problems at scale has motivated me a lot as I take the upcoming steps in my career. I would also like to thank David Kempe, whom I got an opportunity to collaborate with during his time at Cornell.

I am grateful to the students, faculty, and staff of the Department of Computer Science at Cornell for providing me with a community that I am proud to be a part of. Thanks to my officemates – Arzoo Katiyar, Esin Durmus, Tianze Shi, Tobias Schnabel, Yao Cheng – for making Gates 349 a fun and productive place to work. I would like to offer a special thanks to Arzoo Katiyar, who sadly no longer with us anymore, continues to inspire me as a kind, considerate, supportive friend and colleague. I am also thankful to Yi Su, Lequn Wang,

Aman Agarwal, Adith Swaminathan, Angela Zhou, Dipendra Kumar Misra, Ayush Sekhari, Rahul Kidambi for helping me broaden my horizons with our discussions on a wide array of research topics. Thanks also to Marco Morik, Jessica Hong, Hannah Lee, and many other brilliant undergraduate and master's student researchers who I was fortunate to work with and learn from. I also learned a tremendous amount from attending the NLP Seminar and the Machine Learning Discussion Group (MLDG) meetings at Cornell.

My time at Cornell gave me friendships that I will cherish for the rest of my life. An essential aspect of doing a Ph.D. has been about staying mentally and physically fit, especially during the pandemic. I am thankful to some incredible friends who made sure that the dull days became a little more fun. Thanks to Anuj, Arpita, Arzoo, Drishti, Meenakshi, Prateek, Saksham, Utkarsh for every game of squash, tennis, FIFA that we played, and for all the hiking, camping, and road trips together. A special thanks to Rumit Singh for being a big brother, making my troubles go away, and making Ithaca a place I will always call home. I am also glad to have found a great deal of energy and enthusiasm from Jujhar, Ishneet, and the rest of my friends at Cornell Sikh Student Association. Thanks to all of my friends in Ithaca, I fell in love with this gorgeous town, its beautiful seasons, sunrises, and sunsets.

I am blessed to have friends who, even though miles away, have continuously motivated and supported me. I am grateful to Dhruv Lakhani for being a constant friend since our time as next-door neighbors at IIT Kanpur. I always look forward to spending time with Harman, Mohmeet, Tigmanshu, and I fondly remember our adventures in New York City and beyond. Thank you for always inspiring me to strive towards becoming a better person every day. I am also very thankful to my friend Yavnika for the support I gained from shar-

ing our graduate school experiences and making sense of the many highs and lows.

Finally, I am very thankful to my family for their love, sacrifice, and support throughout this long, arduous process. My parents – Rajinder Kaur and Harkirat Singh, my grandparents – Gurbachan Kaur and Mohan Singh, and my sisters – Jasmeet and Jasleen – deserve a lot of credit for everything that I have accomplished. It is because of the sacrifices of my parents from the very beginning that I was gifted with such amazing opportunities. I would also like to thank my relatives – Savinder Singh and Preena Soni – for always welcoming me to their home in California whenever going home to India wasn't an option.

My research was funded in part through NSF Awards IIS-2008139, IIS-1247637, IIS-1217686, IIS-1615706, and IIS-1513692, as well as a gift from Workday Inc. I am very sorry to anyone I forgot to name above; thank you for playing a valuable role in this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Rankings are one of the dominant forms with which online systems present results and recommendations to the user. Far surpassing their conception in library science as a tool for finding books in a library, the prevalence of algorithmic rankings now ranges from search engines and online stores to recommender systems and news feeds. Consequently, it is no longer just books in a library that are being ranked, but there is hardly anything that is not being ranked today – products, jobs, job seekers, rental properties, creative content, opinions, potential romantic partners. Nevertheless, one of the guiding technical principles behind the optimization of algorithmic ranking systems still dates back to more than four decades ago – namely the Probability Ranking Principle (PRP) (Robertson, 1977). It states that the ideal ranking should order items in the decreasing order of their probability of relevance since this is the ranking that maximizes the utility of the retrieval system to the user for a broad range of utility measures in Information Retrieval. In this work, we start with the question – is this uncompromising focus on utility to the users still appropriate when we are not ranking books in a library, but people, products, and opinions?

There are now substantial arguments and precedent that many of the ranking systems in use today have a responsibility not only to their users but also to the items that are being ranked. In particular, the scarce resource that ranking systems allocate is the exposure of items to users, and exposure is largely determined by the position in the ranking – and so is a job candidate's chances to be interviewed by an employer, an Airbnb host's ability to rent out their property, a product to be sold, or a writer to be read. Moreover, given that these platforms act as an arbiter of exposure, the companies operating these platforms are subject to legal and reputation risks. Disagreements on the allocation of exposure have already led to high-profile legal challenges such as the European Union antitrust violation fine on Google (Scott, 2017), and it has sparked a policy debate about search neutrality (Grimmelmann, 2011), and claims of gender and racial discrimination on such platforms (Ferraro et al., 2021).

This dissertation identifies that exposure is the quantity that corresponds to economic opportunity on these online platforms and marketplaces, and focuses on developing a conceptual and computational framework for the fair allocation of exposure to the items being ranked. The two major contributions of this work are: (a) formulating fairness constraints on rankings and (b) providing efficient algorithms for computing utility-maximizing rankings subject to such fairness constraints. In addition to the flexibility of balancing fairness to the items and the utility that the rankings provide to the users, the framework is defined such that we are not limited to a single definition of fairness since different application scenarios probably require different trade-offs between the rights of the items and what can be considered an acceptable loss in utility to the user. Using its expressive power to link exposure, relevance, and impact, a broad range of fairness constraints can be instantiated in this framework. The ranking algorithms developed in this work provide provable guarantees for optimizing expected utility while obeying the specified notion of fairness in expectation.

1.1 Motivation

While it is unlikely that there will be a universal definition of fairness that is appropriate across all applications, let us discuss three different perspectives of how a ranking system may be perceived as unfair or biased in its treatment of the ranked items and where the ranking system may want to impose *fairness constraints* that guarantee some notion of fairness. In particular, these perspectives illustrate how fairness can be related to a biased allocation of opportunity (Section § 1.1.1), misrepresentation of real-world distributions (Section § 1.1.2), and fairness as an instantiation of the freedom of speech principle (Section § 1.1.3), respectively.

1.1.1 Fairly Allocating Economic Opportunity

Many modern applications that use rankings are two-sided markets where both sides of the market derive utility and incentives from the system. In this view, the ranking system serves the users by helping them discover relevant items, and it is the arbiter of exposure from the point of view of the items. To illustrate this point, let us consider three examples where simply optimizing the ranking quality to the users – as done by virtually all current LTR algorithms – leads to unfair or undesirable allocation of exposure among the items.

We start by considering a simple position-based model (PBM) (Chuklin et al., 2015; Joachims et al., 2005; Craswell et al., 2008) of exposure, where the probability that an item will be seen (aka observed) by a user depends only on its rank. The leftmost table in Table 1.1 shows a hypothetical exposure drop-off, where

		(a) Query: *Software Engineer*		(b) Query: *Pop Music*		(c) Query: *Today's news*	
Rank	Exposure	Item	P(interview)	Item	P(like)	Item	P(like)
1	100%	Adam	0.5099	Artist 1	0.5099	Times 1	0.5099
2	90%	Bob	0.5098	Artist 2	0.5098	Times 2	0.5098
3	81%	Charlie	0.5097	Artist 3	0.5097	Times 3	0.5097
...	...	...	...	...	...	...	...
100	0.91%	Alice	0.4999	Artist 100	0.4999	Post 1	0.4999
101	0.90%	Barbara	0.4998	Artist 101	0.4998	Post 2	0.4998
102	0.89%	Claire	0.4997	Artist 102	0.4997	Post 3	0.4997
...	...	...	...	...	...	...	...

Table 1.1: Three examples illustrating how a ranking algorithm that is agnostic to the allocation of exposure to the items can have a profound impact on the fairness properties and dynamics of a marketplace.

the top-ranked item is seen by 100% of the users and the item at position 102 is only seen by 0.89%. For this exposure drop-off curve, consider the following three ranking applications.

Hiring and Discrimination. Table 1.1(a) shows a ranking for the query "Software Engineer" in a hypothetical search engine that lets employers search candidate resumes. Assume that the system defines relevance of a candidate as their probability of being invited for an interview, and perfectly knows the relevance of all job candidates. The ranking shown is the optimal ranking according to the Probability Ranking Principle (PRP), which is known to maximize retrieval quality to the users for virtually all measures of ranking performance (Robertson, 1977). However, the ranking is arguably not fair to the candidates. In particular, Alice is almost as relevant as Adam (one percentage point difference), but Adam receives vastly more exposure than Alice (100% vs. 0.91%). Furthermore, let's say we are given that the candidates in positions 1 to 100 are all males, the candidates in positions 101 to 200 are all females, and that the small difference in "relevance" is due to a small fraction of employers discriminating against female candidates (e.g., 1% in this case). Now, this ranking has an outsized influence on the likelihood of female candidates being discovered and invited for job interviews, greatly amplifying the relatively smaller gender bias in the past data.

Superstar Economics. Table 1.1(b) is analogous to the job-candidate ranking but set in a music streaming service that ranks artists by their relevance to the query "Pop Music". Here relevance reflects whether a user will like a particular artist. Even in this case, the ranking is likely to be perceived as unfair by the artists, and it may have undesirable effects on the health of the market. In particular, since the top few artists receive almost all the exposure, all other artists may get deprived of streaming revenue and driven away from the streaming service. This leads to a market with less high-quality music and "superstar economics" (Rosen, 1981; Mehrotra et al., 2018), where a few participants receive most of the revenue. Moreover, if "likes" are naively used for estimating relevance, this further amplifies the rich-get-richer dynamics (Chapter 5). Overall, the service becomes less competitive, as monopolies drive up the price and users have less choice in music.

Polarization. Table 1.1(c) shows yet another example of the same ranking, but here it corresponds to a hypothetical news-aggregator that would like to present a (non-personalized) "Today's news" ranking on its homepage. Assume that the news-aggregator only has two sources of news – *The Times* and *The Post* – that are politically left-leaning and right-leaning respectively. This news aggregator wants to appeal to all political views to maximize its user base. Currently, let us say the website has an almost balanced user base, where *The Post* get read only slightly more often than *The Times* (~51% vs. ~49%). Unfortunately, ranking by the PRP again leads to undesirable effects, where *The Times*' articles receive far less exposure than they deserve. This is likely to alienate left-leaning users and drive them off the platform. Note that naively enforcing diversity (e.g., enforcing that one article for each viewpoint is represented in the ranking) is not a satisfactory solution since it would overexpose fringe viewpoints or lower-quality content. We argue that it is fundamentally a problem of disconnect between exposure allocation and relevance to the audience.

These examples illustrate how the mechanism by which exposure is allocated to the items can profoundly impact fairness and the dynamics of the marketplace. We may ask the question — is this winner-take-all allocation of exposure fair in these contexts, even if the winner only has a tiny advantage in terms of relevance?[1] Instead, it may seem reasonable to distribute exposure more evenly, even if it leads to a small drop in utility to the user (e.g., the employer, music listener, or reader in the examples above). Even though the examples are numerically equivalent, note that no single fair ranking would be considered fair for all the three scenarios and that the definition of fairness depends on context and application.

1.1.2 Fairly Representing a Distribution of Search Results

Sometimes the results of a query may be used as a statistical sample – either explicitly or implicitly. For example, a user may expect that an image search for the query "CEO" on a search engine returns roughly the right number of male

[1] Note that the tiny advantage may come from a tiny fraction of the gender-biased employers (in the hiring example) or a sampling bias, however, this is not a problem we are addressing here.

Figure 1.1: Image Search example: A sample image search result page for the query "CEO" showing a disproportionate number of male CEOs (Butterly, 2015).

and female executives, reflecting the true distribution of male vs. female CEOs in the world. However, if a search engine returns a highly disproportionate number of males as compared to females (like in the hypothetical search results in Figure 1.1), then the search engine may be perceived as *biased*. In fact, a study detected the presence of gender bias in image search results for a variety of occupations and showed that such biases indeed affect user's beliefs about various occupations (Kay et al., 2015). In other words, a biased information environment may affect users' perceptions and behaviors.

1.1.3 Giving Speakers Fair Access to Willing Listeners

Ranking systems play an increasing role as a medium for speech, creating a connection between bias and fairness in rankings and principles behind freedom of speech (Grimmelmann, 2011). While the ability to produce speech and make this speech available on the internet has undoubtedly created new opportunities to exercise freedom of speech as a speaker, there remains the question of whether or not free speech makes its way to the interested listeners. Hence, the study of the medium becomes necessary. Search engines are the most popular mediums of this kind and therefore have an immense capability of influencing user attention through their editorial policies, which has sparked a policy debate around search neutrality (Introna and Nissenbaum, 2000; Granka, 2010; Grimmelmann, 2011). Similarly, news-aggregators play a similar role in influencing user attention through their recommendation policies, for example, the setup in Table 1.1(c) with a liberal and a conservative news source where a ranking only based on the fraction of the population to who a news article is relevant leads to unfair exposure to one of the news sources, and therefore, raising a similar question about neutrality. While no unified definition of search neutrality exists, many argue that search engines should have no editorial policies other than that their results are comprehensive, impartial, and solely ranked by relevance (Grimmelmann, 2011; Raff, 2009). The question remains whether ranking solely on relevance necessarily means the Probability Ranking Principle, or are there other relevance-based ranking principles that lead to a medium with a more equitable distribution of exposure and access to willing listeners?

The three different perspectives – allocation of economic opportunity, representation of real-world distributions, and fairness as an instance of freedom of speech – indicate how an algorithm that only optimizes for user utility might have undesirable outcomes for the items and their speakers, creators, and suppliers. The examples also demonstrate the drawbacks of the Probability Ranking Principle (PRP), the guiding technical principle behind optimizing ranking systems for more than four decades, motivating us to develop methods that can enforce a desirable relationship between merit and exposure to the items. Since the definition of merit is subjective and depends on the application, it is unlikely that there is a universal definition of fairness that is appropriate across all applications. Therefore, this thesis presents a conceptual and computational framework for formulating fairness constraints on rankings and the associated efficient algorithms for computing utility-maximizing rankings subject to such fairness constraints. A broad range of fairness constraints can be implemented in the framework using its expressive power to link merit, relevance, exposure, and impact. In this way, we are not limited to a single definition of fairness since different application scenarios probably require different trade-offs between the rights of the items and what can be considered an acceptable loss in utility to the user. The next section presents the concrete contributions and the outline of the rest of the thesis.

1.2 Outline and Contributions

We will first review the basics of Information Retrieval (IR) and Learning-to-Rank (LTR) in Chapter 2. Section § 2.2 provides background on different aspects of an LTR algorithm. Section § 2.2.1 reviews different ways in which relevance is defined in the IR literature, while we discuss evaluation measures in Section § 2.2.2 and compare pointwise, pairwise, and listwise approaches for LTR in Section § 2.2.3. To define position bias and exposure in the context of fairness, Section § 2.2.4 talks about user click models and learning from click data. Section § 2.3 surveys four strands of prior work to situate the thesis in the ongoing work in the research community: algorithmic bias and fairness; fairness for rankings; diversified ranking in information retrieval; and bias and discrimination in online systems.

Chapter 3 considers the fairness of rankings through the lens of exposure allocation between groups. Instead of defining a single notion of fairness, we developed a general framework that employs probabilistic rankings and linear programming to compute the utility-maximizing ranking under a whole class of fairness constraints (Section § 3.1). To verify the expressiveness of this class, we showed how to express fairness constraints motivated by the concepts of demographic parity, disparate exposure, and disparate impact (Section § 3.2).

Chapter 4 describes a Learning-to-Rank (LTR) algorithm – named FAIR-PG-RANK – that not only maximizes utility to the users but also rigorously enforces merit-based exposure constraints towards the items. First, we develop a conceptual framework in which it is possible to formulate fair LTR as a policy-learning problem subject to fairness constraints (Section § 4.1.1). We show that viewing fair LTR as learning a stochastic ranking policy leads to a rigorous formulation that can be addressed via Empirical Risk Minimization (ERM) on both the utility and the fairness constraint. Second, we propose a class of fairness constraints for ranking that incorporates notions of both individual and group fairness (Section § 4.1.2). Finally, we propose a policy-gradient method for implementing the ERM procedure that can directly optimize any information retrieval utility metric and a wide range of fairness criteria (Section § 4.2.2.1). Across a number of empirical evaluations, we also find that the policy-gradient approach is a competitive LTR method in its own right (Section § 4.3.2). We also show that FAIR-PG-RANK can identify and avoid biased features when trading-off utility for fairness (Section § 4.3.3), and that it can effectively optimize notions of individual and group fairness on real-world datasets (Section § 4.3.4).

In a real-world scenario, the data is often revealed to the training algorithm one query at a time, and even then, it might be incomplete because the user does not examine all the documents in the ranking. In Chapter 5, we tackle the problem of learning a ranking function that ensures notions of amortized group fairness while simultaneously learning the ranking function from implicit feedback data. Section § 5.4 formally introduces the *dynamic learning-to-rank* setting, followed by Section § 5.5 that formalizes an amortized notion of merit-based fairness, accounting for the fact that merit itself is unknown at the beginning of the learning process and is learned throughout. Section § 5.6 then addresses the bias problem, providing estimators that eliminate the presentation bias for both global and personalized ranking policies. Finally, Section § 5.7 proposes a control-based algorithm that is designed to optimize ranking quality while dynamically enforcing fairness. It does so by integrating unbiased estimators for both fairness and utility, dynamically adapting both as more data becomes available. In addition to its rigorous theoretical foundation and convergence guarantees, we find empirically that the algorithm is highly practical and robust (Section § 5.8).

Finally, Chapter 6 describes an axiomatic framework to define fairness in ranking in the presence of uncertainty of merit. The framework lends itself to an algorithm to obtain (approximately) fair ranking distributions (Section § 6.3). This framework, including extensions to approximate notions of fairness, is presented in Section § 6.2. We also illustrate that uncertainty of merit is an important source of unfairness, and modeling it explicitly and axiomatically is key to addressing it.

Chapter 7 concludes this thesis and provides future research directions while discussing challenges in the area that need to be addressed.

1.2.1 Bibliographical Note

The following publications form the basis of the chapters in this thesis. Chapter 3 is based on joint work with Thorsten Joachims, published in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'18) (Singh and Joachims, 2018). Chapter 4 is based on joint work with Thorsten Joachims, published in Advances in Neural Information Processing Systems 32 (NeurIPS 2019) (Singh and Joachims, 2019). Chapter 5 is based on joint work with Marco Morik, Jessica Hong, and Thorsten Joachims, published in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval 2020 (Morik et al., 2020). Marco Morik and I are the joint first authors of the publication and contributed to the work equally. Chapter 6 is based on joint work with David Kempe and Thorsten Joachims (Singh et al., 2021).

CHAPTER 2

BACKGROUND

This chapter introduces the concepts and related work that this work builds upon. Section § 2.1 gives an introduction to the area of information retrieval. Section § 2.2 describes the learning-to-rank framework, different evaluation measures and discusses different approaches from the literature. Furthermore, section § 2.2.4 discusses the challenges and existing approaches for learning from user clicks and also reviews proposed click models for modeling human behavior. Section § 2.3 reviews relevant research in the areas of algorithmic fairness, including recent work on fairness in ranking, diversity in information retrieval, and bias in online platforms.

2.1 Information Retrieval: Background

The goal of an information retrieval (IR) system is to provide users with access to information that is typically stored as a collection of documents. The user initiates an interaction with the system by issuing a query (usually a sequence of words but could refer to another form through which the user expresses the information need to the system). The task of the system is to select information (e.g., an entire document or a specific part of a document) to present to the user that is most likely to be *relevant* to the user's information need. The concept of a document being *relevant* to the query and the user's information need is a central topic of study in the area of information retrieval, and we later discuss it in Section § 2.2.1. This retrieved information is most often presented to the user as a ranked list of documents.

The research in the area of information retrieval systems to learn effective retrieval functions can be categorized into two categories broadly, one focusing on representing the relationship between the queries and the documents, and the other focusing on the techniques used to define, train and fine-tune the ranking function that defines how a candidate set of documents are presented to the user. However, modern IR systems, especially those using deep learning techniques, usually do not have a clear boundary between the representation and the ranking functions, but we will review the two categories separately for a more straightforward exposition. Later in this work, we focus mostly on the ranking function – its properties and the methods to learn it, assuming that the query and document representations are already known.

In early retrieval systems, a boolean model was used to represent the presence or absence of information in documents, and the queries could be formulated as logical clauses to form a retrieval function (Salton and McGill, 1983). These systems were improved upon by systems that weighed different terms in a document differently to develop a notion of relevance. These further led to the development of a vector space model where each document and query could be represented as a point in vector space (Salton et al., 1975; Salton and McGill, 1983). Some very recent developments in representation learning that use deep neural networks for document representation (see Mitra et al. (2018) for an overview), for example, BERT (Khattab and Zaharia, 2020; Nogueira and Cho, 2019), have shown significant improvements in the efficacy of IR systems.

The concept of rankings was formalized as the Probability Ranking Principle (PRP) by Robertson (1977), which states that a ranking that sorts documents by their probability of relevance to the user maximizes the utility to the user. PRP originated in library science, where the primary aim of the search functionality was to retrieve a set of books from a library catalog that best matched the user's query. As discussed below and in more detail in Chapter 3, for most common utility functions developed to evaluate the quality of an IR system, PRP refers to the optimal ranking function. Hence, most research has focused on accurately predicting the probability of relevance for query-document pairs or the relative ordering of the documents with respect to their probabilities of relevance.

2.1.1 Note on Recommender Systems

In this work, even though we will primarily consider the setup of information retrieval described above, note that recommender systems may be considered a particular case of an IR system that works without a query issued by a user (Belkin and Croft, 1992). Instead of a query, a recommender system considers the user profile (e.g., user's history of interactions with the system through explicit and implicit feedback) and other contextual information (e.g., time of day, location) to identify the most relevant information for the user. On the other hand, this use of contextual information to broaden the meaning of a query corresponds to the area of personalization in IR (Fan and Poole, 2006).

Another aspect that makes a recommender system different from an IR system is the treatment of this information, i.e., the documents and items. Traditionally, research in IR has focused on text-based retrieval, i.e., both the query and the documents are textual, and hence there is much attention given to the methods for text representation. On the other hand, research in recommender systems often does not assume that a content-based representation of the items exists. In this way, specialized recommender systems such as content-based recommender systems (for example, a recommender system for scholarly articles or news) may be considered similar to text-based retrieval systems. In short, even though the two areas of Information Retrieval and Recommender Systems have been studied differently over the past few decades, they are closely related and many techniques transfer across.

In this work, since we primarily consider the task of ranking, we can consider the IR setup for recommender systems as well, where we already have a way to represent a user's history in the same way as we would represent a query. The following section describes the Learning-to-Rank framework for an information retrieval-based setup.

2.2 Learning-to-Rank

Learning-to-Rank (LTR) is a machine learning task[1] where the goal is to *learn* a ranking function that orders a given set of documents based on a query. As described above, a learning-to-rank algorithm is provided with a set of queries, a corresponding set of documents, and expert relevance judgments for the document in the set. These relevance judgments are typically available in the form of a numerical or ordinal score, or a binary judgment (0: relevant, 1: irrelevant) for each document.

The goal of a learning-to-rank algorithm is to optimize the quality of the ranking function as measured by one or more utility metrics (e.g., DCG, MAP,

[1] Depending on the format of training data, it might be considered as a supervised, semi-supervised, or reinforcement learning task.

Precision@k) on a test dataset.

2.2.1 Relevance

As described above, relevance judgments for a set of documents with respect to a query may either be known in the form of a binary relevance judgment (0: irrelevant, 1: relevant), an ordinal score (e.g., Likert scale in product ratings), or a numerical score (e.g., the fraction of users issuing query who will find the document relevant).

The concept of relevance is central to IR research and especially this thesis since we define a framework that explicitly ties together relevance with the position in the ranking (through exposure described later in § 2.2.4.1). Despite being at the center of much of the research in the field, the meaning of "relevance" has been debated throughout the history of the field. According to Saracevic (1996), relevance can be operationalized in several ways, topical relevance (whether a document is related to a given topic), cognitive relevance (concerning the relationship between the presented information and the user's cognitive state, e.g., background or domain knowledge), situational relevance (concerning the relationship between the retrieved document and user's situation, e.g., task completion), and motivational relevance (relation between intents, goals, and motivations of a user and the retrieved document, e.g., satisfaction).

2.2.1.1 Merit

Even though the framework and algorithms proposed in this work are agnostic to the specific operationalization of relevance, to define any fairness notion based on relevance, we need to connect relevance with merit. In this work, this connection is defined using an arbitrary monotonic function f that maps relevance to merit as

$$\text{Merit} = f(\text{Relevance}).$$

While in the case of binary relevances, the function f may correspond to an identity function, i.e., relevant items are meritorious while non-relevant items are not, for numerical or ordinal relevances, the specific choice of function f would be application-dependent. Chapter 3 presents a discussion on this connection between merit and relevance.

2.2.2 Evaluation Measures: Utility functions

The traditional goal of an IR system is to maximize the utility of the system for the user. Since utility as an economic quantity might be harder to define, a common way used to express utility in IR is to evaluate the ranking function based on expert relevance judgments. Since the relevance judgments might in the form of a binary relevance judgment or a numerical or ordinal score, several metrics work under these different forms. Some example evaluation measures are Mean Average Precision (MAP), Discounted Cumulative Gain (DCG) and Normalized Discounted Cumulative Gain (NDCG), Precision@k, Mean Reciprocal Rank (MRR), Kendall's tau. While DCG and NDCG are usually preferred when graded relevances are used, metrics such as MAP, precision, and MRR are only suitable for binary relevance judgments (see for a survey). In this work, we will primarily use the example of DCG and NDCG as they are widely used in both academic and industry applications of Learning-to-Rank, but all of our methods translate to other evaluation metrics that are decomposable into a sum over different documents in the ranking (more details in Section § 3.1.1).

However, for the purposes of optimization, an LTR algorithm may choose to optimize a different loss function. Based on the way an LTR algorithm sets up its optimization problem, it can be categorized into three approaches described below.

2.2.3 Pointwise, Pairwise, Listwise Learning-to-Rank

Based on the difference in terms of either the input space, output space, or the hypothesis space as well as the loss functions, Learning-to-rank methods can broadly be categorized into three approaches, namely *pointwise*, *pairwise*, and *listwise* methods (Liu et al., 2009).

Pointwise learning-to-rank. In the case of pointwise learning-to-rank, the algorithm assumes that each query-document pair is associated with a ground truth relevance label. Then the learning-to-rank problem can be solved as a classification problem in the case of binary labels, or as a regression problem in the case of numerical scores (Nallapati, 2004; Cossock and Zhang, 2006). At the prediction time, the ranking is essentially a list of documents sorted by their predictions, i.e., the predicted probability of the document to be relevant to the query or the predicted numerical score. In the case of ordinal labels, methods such as ordinal regression have been proposed that simultaneously learn a mapping to output scores and thresholds for different relevance labels (Crammer and Singer, 2002).

Pairwise learning-to-rank. In the pairwise approach, the problem is reduced to a classification problem of learning a binary classifier that compares two documents with respect to a query to predict whether one is more relevant than the other. The 0-1 loss of this binary classification problem reflects the number of inversions in the ranking. In contrast to the pointwise approach, pairwise learning-to-rank has a higher complexity since it is quadratic in the number of documents.

Also, using a loss function based on the binary classification loss may have a significant mismatch with IR evaluation measures that are more sensitive to the top of the ranking as compared to the bottom.

A widely known and effective approach using pairwise learning-to-rank, called SVM-Rank, uses a support vector machine (SVM) approach to minimizing a pairwise hinge loss (Herbrich, 1999; Joachims, 2002, 2006).

Listwise learning-to-rank. Listwise approaches try to directly optimize the value of one of the evaluation measures in Section §2.2.2 averaged over all queries in the training data. However, since most of the evaluation measures are not continuous functions with respect to the model parameter, methods use smooth approximations or bounds for the purposes of optimization. Some notable examples are SoftRank (Taylor et al., 2008), ListNet (Cao et al., 2007), LambdaRank (Burges et al., 2005), and LambdaMART (Burges, 2010).

2.2.4 Learning to rank from user clicks

An expert relevance judgment is an example of explicit feedback. Although an expert judgment is a reliable source of relevance as it is provided by trained experts who are acting consciously, it is extremely expensive to collect at scale. Meanwhile, implicit user feedback, such as clicks, is very cheap for the system to obtain as it is the by-product of a user's natural interaction with the system. However, a consequence of implicit feedback not being a conscious relevance judgment is that it is often noisier than explicit feedback (Fox et al., 2005), and therefore, using clicks to train a learning-to-rank algorithm raises the question of how to interpret user behavior for learning correctly (Joachims et al., 2005; Carterette and Jones, 2007).

Position Bias. A key challenge in using click feedback is that clicks only provide a meaningful relevance signal for the documents that are examined by the user. Meanwhile, the documents in the ranking that the user does not examine are never clicked even if they are relevant to the user's information need. For example, documents placed higher in the ranking while collecting the feedback are more likely to get clicked as compared to equally relevant documents placed lower in the ranking, and hence using clicks and non-clicks directly as binary relevance judgments could be biased by position (Joachims, 2002; Joachims et al., 2017). To accurately interpret these clicks, user click models have been developed (discussed below) that use a generative model of user behavior to explain the observed clickthrough data.

2.2.4.1 User Click Models

A user click model, intuitively, is a set of rules that allows us to simulate user behavior on a ranking in the form of a random process. An important motivation for developing click models has been that they help in reasoning about user behavior when it is not possible to experiment with real users. Apart from simulating user behavior, click models are also useful to improve ranking functions (e.g., by inferring relevance from clicks predicted by a click model) and for developing better evaluation metrics (e.g., model-based metrics).

A position-based model (PBM) is based on the examination hypothesis which states that a user clicks a document if and only if they examine the document and are *attracted* by it (Craswell et al., 2008; Joachims et al., 2005). The idea of *attractiveness* of a document is different from relevance because it is a characteristic of the document's visible snippet or caption and not the full document. Joachims et al. (2005) show that the probability of a user examining a document depends heavily on its position in the ranking. Hence, the PBM (Craswell et al., 2008) defines the probability of a click as the product of the probability of the user examining the document which only depends on the rank of the document, and the probability of the user being attracted to the document. In this work, we will not distinguish between the relevance and attractiveness of a document, but one may consider alternative models that consider this relationship as well.

A cascade model (CM) assumes that a user examines documents in a ranking from top to bottom until they find a relevant document (Craswell et al., 2008). The model assumes that the top-ranked document is always examined and the subsequent documents are examined if and only if the previous documents were examined and not clicked. A CM is different from a PBM in two ways: a CM cannot be used for settings where multiple clicks are observed on a single ranking, and that in PBM, the probability of clicking a document is independent of the relevance of the previous documents in the ranking and only depends on its own relevance and position. Furthermore, user browsing model (UBM) (Dupret and Piwowarski, 2008) and dependent click model (DCM) (Guo et al., 2009) provide extensions to a cascade model meant to handle multiple clicks on a ranking. Chuklin et al. (2015) provides a detailed survey on UBM, DCM, and other click models as well.

Position Bias and Exposure. The position bias at position k is defined as the probability that a user who views the ranking will examine the item ranked at position k. This quantity can be defined for each ranking, or it could be defined as a marginal probability over different rankings under the same query. The position bias captures how much attention an item will receive, where higher positions are expected to receive more attention than lower positions. Under a position-based model (PBM), the position bias at position k is only a function of k, while for other click models, it may depend on the collection of items and the distribution of queries. In operational systems, position bias can be directly measured using eye-tracking (Joachims et al., 2007), or indirectly estimated through swap experiments (Joachims et al., 2017), or intervention harvesting to harness natural experiments occurring in observational data (Agarwal et al., 2019b; Fang et al., 2019).

Meanwhile, the *exposure* of a document is defined as the expected amount of attention a document receives. In other words, it is the position bias at the position where the document is placed in the ranking.

2.3 Related Work

This section surveys four directions of prior work to situate the thesis in the ongoing work in the research community. First, this thesis draws on concepts for algorithmic fairness for supervised learning in the presence of sensitive attributes (Section § 2.3.1). Second, Section § 2.3.2 presents prior work on existing and parallel work on algorithmic fairness for rankings. Third, this section contrasts the notions of fairness proposed in this work with the problem of fair division in economics and game theory (Section § 2.3.3), and the well-studied areas of diversified ranking in information retrieval (Section § 2.3.4). Finally, it presents related works that examine how bias manifests in online systems such as advertisement platforms and web search (Section § 2.3.5).

2.3.1 Algorithmic Bias and Fairness

As algorithmic techniques, especially machine learning, find widespread applications, there is much interest in understanding its societal impacts. While algorithmic decisions can counteract existing biases, algorithmic and data-driven decision-making affords new mechanisms for introducing unintended bias (Barocas and Selbst, 2016). There have been numerous attempts to define notions of fairness in the supervised learning setting. The individual fairness perspective states that two individuals similar with respect to a task should be classified similarly (Dwork et al., 2012). Individual fairness is hard to define precisely because of the lack of agreement on task-specific similarity metrics for individuals. There is also a group-fairness perspective for supervised learning that implies constraints like demographic parity and equalized odds. Demographic parity posits that decisions should be balanced around a sensitive attribute like gender or race (Calders et al., 2009; Zliobaite, 2015). However, it has been shown that demographic parity causes a loss in the utility and infringes individual fairness (Dwork et al., 2012) since even a perfect predictor typically does not achieve demographic parity. The equalized odds definition represents the equal opportunity principle for supervised learning and defines the constraint that the false positive and true positive rates should be equal for different protected groups (Hardt et al., 2016). While not all of these definitions are compatible with each other (Kleinberg et al., 2017; Chouldechova, 2017), there is significant interest in developing notions of fairness specifically suited for different supervised learning tasks (Barocas et al., 2019). Several recent works have also focused on learning algorithms compatible with these definitions of fair classification (Zemel et al., 2013; Woodworth et al., 2017; Zafar et al., 2017), including causal approaches to fairness (Kilbertus et al., 2017; Kusner et al., 2017; Nabi and Shpitser, 2018).

2.3.2 Fairness in Rankings

Rankings are a primary interface through which machine learning models support human decision-making, ranging from recommendation and search in online systems to machine-learned assessments for college admissions and recruiting. One added difficulty when considering fairness in the context of rankings is that the decision for an agent (where to rank that agent) depends not only on their own merits but on others' merits as well (Dwork et al., 2019). The existing work can be roughly categorized into three groups: *Composition-based*, *Opportunity-based*, and *Evidence-based* notions of fairness.

The notions of fairness based on the composition of the ranking operate along the lines of demographic parity (Zliobaite, 2015; Calders et al., 2009), proposing definitions and methods that minimize the difference in the (weighted) representation between groups in a prefix of the ranking (Yang and Stoyanovich, 2017; Celis et al., 2018; Asudeh et al., 2019; Zehlike et al., 2017; Mehrotra et al., 2018; Zehlike and Castillo, 2020). Yang and Stoyanovich (2017) propose statistical parity-based measures that compute the difference in the distribution of different groups for different prefixes of the ranking (top-10, top-20, and so forth). The differences are then averaged for these prefixes using a discounted weighting (like in evaluation measures such as DCG). This measure is then used as a regularization term for a ranking algorithm. Zehlike et al. (2017) formulate the problem of finding a 'Fair Top-k ranking' that optimizes utility while satisfying two sets of constraints: first, in-group monotonicity for utility (i.e., more relevant items above less relevant within the group), and second, a fairness constraint that the proportion of protected group items in every prefix of the *top-k* ranking is above a minimum threshold. Celis et al. (2018) propose a constrained maximum weight matching algorithm for ranking a set of items efficiently under a fairness constraint indicating the maximum number of items with each sensitive attribute allowed in the top positions. Some recent approaches, like Asudeh et al. (2019), have also looked at the task of designing fair scoring functions that satisfy desirable fairness constraints.

Other works argue against the winner-take-all allocation of economic opportunity (e.g., exposure, clickthrough rates) to the ranked agents or groups of agents, and that the allocation should be based on a notion of merit (Singh and Joachims, 2017, 2018; Biega et al., 2018; Diaz et al., 2020). This is the motivating factor of this thesis as well, and we discuss it in greater depth in the following chapters.

Meanwhile, the metric-based notions equate a ranking with a set of pairwise comparisons, and define fairness notions based on parity of pairwise metrics within and across groups (Kallus and Zhou, 2019; Beutel et al., 2019; Narasimhan et al., 2020; Lahoti et al., 2019). Similar to pairwise accuracy definitions, evidence-based notions such as Dwork et al. (2019) propose semantic notions such as *domination-compatibility* and *evidence-consistency*, based on relative ordering of subsets within the training data.

The fairness notions proposed in this thesis combine the opportunity-based and evidence-based notions by stating that the economic opportunity allocated to the agents must be consistent with their merit (Chapter 3), and merit should be estimated consistently with respect to the existing evidence about the relevance of the items to achieve high utility out of the system (Chapters 4, 5, 6).

While most of the fairness constraints defined in the previous work reflect specific parity constraints restricting the fraction of items with each attribute in the ranking, this work proposes a general algorithmic framework for efficiently computing optimal probabilistic rankings for a large class of possible fairness constraints, and these parity constraints can be instantiated inside the proposed framework.

Groups of ranked items In this work, the broader meaning of protected groups is used based on the application domain. However, work on algorithmic fairness mostly considers groups of people that are protected from discrimination by law, based on sex, race, age, disability, color, creed, national origin, or religion. While in many applications, the definition of a group of ranked items could use these protected attributes (e.g., gender in the job-candidate ranking in Section § 1.1.1), in other applications, it may make sense to use a definition such as a group of items sold by an individual seller or music by an individual artist, as the definition of a group of items. Moreover, as we discuss later, the group-based fairness definitions also extend to individual fairness by considering groups of size one.

2.3.2.1 Multi-stakeholder perspective of Recommender Systems

In the domain of recommender systems, multiple works are congruent with this thesis in calling for a multi-stakeholder perspective of recommender systems that moves beyond user-centric utility maximization (Abdollahpouri et al., 2019; Burke et al., 2018). This perspective leads to considering both fairness for users of the system (Yao and Huang, 2017; Xiao et al., 2017), as well as for the producers (e.g., merchants, artists, job seekers, etc.) (Ekstrand et al., 2018; Beutel et al., 2019; Narasimhan et al., 2020). From the perspective of item popularity (Celma and Cano, 2008; Fleder and Hosanagar, 2009), the rich-get-richer effect due to recommendations leads to a large disparity in the allocation of economic benefit which can also be understood from the principle of fair exposure. In this context, fairness has been defined in terms of pairwise accuracy in inter-group and intra-group comparisons between pairs of items (Beutel et al., 2019; Narasimhan et al., 2020; Kallus and Zhou, 2019). Since pairwise preferences imply rankings on the set of items, one can show that these as well are a special case of our fairness of exposure framework defined in this work.

2.3.3 Fair Division

One key aspect of fair rankings is the fair division of user attention among the items. The problem of fair division has been studied in the intersection of economics, game theory, and mathematics for decades, where the goal is to divide a set of resources among several agents (Varian, 1987; Lang and Rothe, 2016; Moulin, 2003; Brams and Taylor, 1996). In the general setup, each agent has a value function that assigns value to different parts of the resource. The resource can be a set of goods that is either divisible (e.g., money) or indivisible (e.g., a house), and its composition can be homogeneous (e.g., money) or heterogeneous (e.g., a cake with different ingredients in different parts).

Fair cake-cutting is a special case of fair division when dealing with a divisible, heterogeneous good. Some of the classic desiderata for fair divisions are (a) proportionality, i.e., every agent receives their "fair share" of the utility, (b) envy-freeness, i.e., no agent wishes to swap her allocation with another agent, (c) equitability, i.e., all agents have the same value for their allocations. When two agents are involved in a cake-cutting problem, the well-known solution of divide-and-choose is long understood to be fair. However, when extending this to more than two agents, algorithms like the Selfridge-Conway procedure become too complex even for a small number of agents (Su, 2000; Brams and Taylor, 1995). Most of the research in this area has focused on designing protocols for achieving proportionality, envy-freeness, and equitability, or on the design of approximation algorithms in settings where fulfilling the fairness objective exactly is impossible.

In the case of rankings, if we consider the case of allocating exposure to items in an online marketplace, the resource for contention is exposure, and exposure is a homogeneous resource, where the top of the ranking is the most valued, then the second, and so on. Since not all agents (e.g., merchants in a marketplace) deserve equal shares, it makes rankings a special case of fair division called Proportional cake-cutting with different entitlements (Robertson and Webb, 1998). Besides being unequally entitled to exposure, each agent seeks more exposure than they would receive, making it tricky to define envy-freeness.

2.3.4 Information Diversity in Retrieval

Ranking with novelty and diversity has long been an interest in the field of information retrieval. Diversity is usually defined as a goal for rankings to include different subtopics or cover different possible intents of a query (Carbonell and Goldstein, 1998; Carterette, 2011; Zhu et al., 2007).

At first glance, diversity in rankings may appear related to the concept of fairness in rankings, since they both lead to more diverse rankings. However, their motivation and mechanisms are fundamentally different. Like the Probability Ranking Principle, diversified ranking is entirely beholden to maximizing utility to the user, while our approach to fairness balances the needs of users and items. In particular, while both the PRP and diversified ranking maximize utility for the user alone, their difference lies merely in the utility measure that is maximized. Under extrinsic diversity (Radlinski et al., 2009), the utility measure accounts for uncertainty and diminishing returns from multiple relevant results (Carbonell and Goldstein, 1998; Radlinski et al., 2008). Under intrinsic diversity (Radlinski et al., 2009), the utility measure considers rankings as portfolios and reflects redundancy (Clarke et al., 2008). Under exploration diversity (Radlinski et al., 2009), the aim is to maximize utility to the user in the long term through more effective learning.

On the other hand, the work on fairness presented in this dissertation is fundamentally different in its motivation and mechanism, as it does not modify the utility measure for the user but instead introduces rights of the items that are being ranked. Moreover, the novel utility metrics proposed for diversified retrieval may still be compatible with the framework presented in this work.

2.3.5 Bias and Discrimination in Online Platforms

Platforms such as social media, online marketplaces, and news recommenders are designed to maximize their profits. In the case of social media, maximizing profit is analogous to optimizing for user engagement which often comes at the cost of manipulating user attention (Ferrara and Yang, 2015). Moreover, there is no guarantee that profit-maximizing algorithms or mechanisms do not discriminate between users with different sensitive attributes in terms of the opportunity they provide or the impact they have on the users. Meanwhile, only some attention has been given to how discrimination based on protected attributes may further be propagated by online platforms (Baeza-Yates, 2018).

Online Platforms with a Reputation System

Most online marketplaces would not exist without a seller-buyer reputation system that allows buyers to rate sellers and vice versa. These ratings are often used by the underlying algorithms that decide how to rank competing sellers to the buyers using the system. While these reputation systems act as a medium of trust on the platform, it has been argued that they also present several challenges to the marketplace. Besides challenges such as incentives to manipulate ratings and placing a barrier of entry for new sellers, the design may also facilitate discrimination, for instance, examples of discrimination against African-American hosts and guests on Airbnb (Edelman and Luca, 2014; Edelman et al., 2017), drivers on Uber (Rosenblat et al., 2017), sellers on Craigslist and eBay (Nunley et al., 2011). Hence, the ranking and recommendation algorithms based on explicit user feedback signals often used as a proxy for user relevance may further propagate unfairness.

Advertisement Delivery Systems

Similarly, in online advertisement delivery, several studies show how algorithms are responsible for skewing ad delivery towards or away from protected groups. An audit by Sweeney (2013) shows that ads for public records *suggest* arrest records when searching for *black-identifying* names. In a study with job advertisements, Datta et al. (2015) find that female users are shown fewer instances of high-paying jobs as compared to similar male users. More recently, Ali et al. (2019) show that Facebook's ad-delivery system algorithmically skews the reach of an employment or housing ad based on the gender and race of the users despite an advertiser choosing neutral parameters for ad targeting.

Trust Bias

Further, it has been demonstrated that the position assigned to an item affects the user's decision to click because of the user's trust on the ranking algorithm or the entire system (Pan et al., 2007; Joachims et al., 2005). This inherent trust bias towards a search engine, social media, or a news recommender demonstrates their implications on culture, society, and market through the influence on both user attention and web traffic (Gillespie, 2014).

Search Neutrality

More recently, the case of neutrality in search engines has been argued from two perspectives: one that focuses on users and the other that focuses on websites that are being ranked (Grimmelmann, 2011). The users' perspective is the dual of the websites' perspective, and in this work, we will mostly use the websites' perspective as the working principle. However, it is easy to translate the concepts since a website's access interest is a derivative of its users' interest. Based on the interest in search neutrality, various works have argued for the need to regulate search engines, social media, and recommendation systems (Nunziato, 2009).

The "right to reach an audience" for websites (Chandler, 2006) proposes that website owners should be given protection against demotion by search engines that might make them virtually inaccessible on the internet. Such ideas have also gained attention in the news because of a case concerning Google and websites like Foundem, a price comparison website, who found themselves losing business owing to a drop in Google search rankings because of a penalty levied on them (Raff, 2009).

While this work does not operationalize these perspectives of search neutrality, the exposure-allocation framework proposed in this work relates to and may be useful in translating these principles into practice. This thesis argues that, while maximizing engagement and profit through *relevance* are meaningful goals, there is a need to consider fairness aspects of such systems from the two-sided perspective, i.e., users and items, and a way to achieve fairness is by explicitly connecting relevance to exposure since exposure is often the quantity under contention.

2.3.6 Fairness in Real-World Information Systems

This thesis advances the understanding of the impact of ranking systems and formulates notions of fairness for rankings, which has led to a subsequent interest from the industry in rethinking production ranking algorithms. LinkedIn has published improvements focused on improving the fairness properties in multiple products, like LinkedIn's Talent Search functionality (Geyik et al., 2019), and their recommender system recommending network connections (Nandy et al., 2021). Meanwhile, Spotify performed a counterfactual analysis of fairness-focussed interventions in their recommender system by evaluating the trade-off between satisfaction to the users and fairness of representation of artists (Mehrotra et al., 2018). Some research from Google also shows improvements in terms of a pairwise notion of ranking fairness (corresponding to a special case of the Disparate Exposure constraint in Section § 3.2.2) on a large-scale production recommender system (Beutel et al., 2019).

CHAPTER 3

FAIR ALLOCATION OF EXPOSURE IN RANKINGS

Acknowledging the ubiquity of rankings across applications, we conjecture that there is no single definition of what constitutes a fair ranking, and it depends on context and application. In particular, in this chapter[1], we will see that different notions of fairness imply different trade-offs in utility, which may be acceptable in one situation but not in the other. To address this range of possible fairness constraints, Section § 3.1 develops a framework for formulating fairness constraints on rankings, and then computing the utility-maximizing ranking subject to these fairness constraints with provable guarantees. Section § 3.2 utilizes the framework to instantiate fairness constraints that correspond to relevant concepts from the algorithmic fairness literature and provide efficient ranking algorithms with provable fairness guarantees. Finally, Section § 3.2.4 discusses how the framework can further be extended to individual fairness constraints, estimated relevances, and computing cost of fairness.

3.1 A Framework for Ranking under Fairness Constraints

For simplicity, consider a single query q and assume that we want to present a ranking r of a set of documents $\mathcal{D} = \{d_1, d_2, d_3 \ldots, d_N\}$. Denoting the utility of a ranking r for query q with $\mathrm{U}(r|q)$, the problem of optimal ranking under fairness constraints can be formulated as the following optimization problem:

$$\begin{aligned} r &= \operatorname{argmax}_r \mathrm{U}(r|q) \\ &\quad \text{s.t. } r \text{ is fair} \end{aligned}$$

[1] This chapter is based on joint work with Thorsten Joachims (Singh and Joachims, 2018)

In this way, we generalize the goal of the Probabilistic Ranking Principle, which emerges as the special case of no fairness constraints. To fully instantiate and solve this optimization problem, we will specify the following four components. First, we define a general class of utility measures $\mathrm{U}(r|q)$ that contains many commonly used ranking metrics. Second, we address the problem of how to optimize over rankings, which are discrete combinatorial objects, by extending the class of rankings to probabilistic rankings. Third, we reformulate the optimization problem as an efficiently solvable linear program, which implies a convenient yet expressive language for formulating fairness constraints. Finally, we show how a probabilistic ranking can be recovered from the solution of the linear program efficiently.

3.1.1 Utility of a Ranking

Virtually all utility measures used for ranking evaluation derive the utility of the ranking from the relevance of the individual items being ranked. For each user u and query q, $\mathrm{rel}(d|u,q)$ denotes the binary relevance of the document d, i.e., whether the document is relevant to user u or not. Note that different users can have different $\mathrm{rel}(d|u,q)$ even if they share the same q. To account for personalization, we assume that the query q also contains any personalization features and that $\mathcal{U}$ is the set of all users that lead to identical q. Beyond binary relevance, rel could also represent other relevance rating systems such as a Likert scale in movie ratings or a real-valued score.

A generic way to express many utility measures commonly used in information retrieval (Section § 2.3.5) is

$$\mathrm{U}(r|q) = \sum_{u \in \mathcal{U}} P(u|q) \sum_{d \in \mathcal{D}} v(\mathrm{rank}(d|r)) \lambda(\mathrm{rel}(d|u,q)),$$

where v and λ are two application-dependent functions. The function $v(\mathrm{rank}(d|r))$ models how much attention document d gets at rank $\mathrm{rank}(d|r)$, and λ is a function that maps the relevance of the document for a user to its utility. In particular, the choice of v could be based on the position bias i.e., the fraction of users who examine the document shown at a particular position out of the total number of users who issue the query q. The choice of λ mapping relevance to utility is somewhat arbitrary[2]. For example, a widely used evaluation measure, Discounted Cumulative Gain (DCG) (Järvelin and Kekäläinen, 2002) can be represented in our framework where $v(\mathrm{rank}(d|r)) = \frac{1}{\log(1+\mathrm{rank}(d|r))}$, and $\lambda(\mathrm{rel}(d|u,q)) = 2^{\mathrm{rel}(d|u,q)} - 1$ (or sometimes simply $\mathrm{rel}(d|u,q)$):

$$\mathrm{DCG}(r|q) = \sum_{u \in \mathcal{U}} P(u|q) \sum_{d \in \mathcal{D}} \frac{2^{\mathrm{rel}(d|u,q)} - 1}{\log(1 + \mathrm{rank}(d|r))}$$

For a measure like $DCG@k(r|q)$, we can choose $v(\mathrm{rank}(d|r)) = \frac{1}{\log(1+\mathrm{rank}(d|r))}$ for $\mathrm{rank}(d|r) \leq k$ and $v(\mathrm{rank}(d|r)) = 0$ for $\mathrm{rank}(d|r) > k$.

Since utility is linear in both v and λ, we can combine the individual utilities into an expectation

$$\begin{aligned} \mathrm{U}(r|q) &= \sum_{d \in \mathcal{D}} v(\mathrm{rank}(d|r)) \Big(\sum_{u \in \mathcal{U}} \lambda(rel(d|u,q))\, P(u|q) \Big) \\ &= \sum_{d \in \mathcal{D}} v(\mathrm{rank}(d|r)) u(d|q), \end{aligned}$$

where

$$u(d|q) = \sum_{u \in \mathcal{U}} \lambda(rel(d|u,q))\, P(u|q)$$

[2] We will refer to the per-item utility as the merit of the item in the subsequent chapters of this thesis.

is the expected utility of a document d for query q. In the case of binary relevances and λ as the identity function, $u(d|q)$ is equivalent to the probability of relevance. It is easy to see that sorting the documents by $u(d|q)$ leads to the ranking that maximizes the utility

$$\operatorname{argmax}_r \mathrm{U}(r|q) \equiv \operatorname*{argsort}_{d \in \mathcal{D}} u(d|q)$$

for any function v that decreases with rank. This is the insight behind the Probability Ranking Principle (PRP) (Robertson, 1977).

3.1.2 Probabilistic Rankings

Rankings are combinatorial objects, such that naively searching the space of all rankings for a utility-maximizing ranking under fairness constraints would take time that is exponential in $|\mathcal{D}|$. To avoid such combinatorial optimization, we consider probabilistic rankings R instead of a single deterministic ranking r. A probabilistic ranking R is a distribution over rankings, and we can naturally extend the definition of utility to probabilistic rankings as follows.

$$\begin{aligned} \mathrm{U}(R|q) &= \sum_{r} R(r) \sum_{u \in \mathcal{U}} P(u|q) \sum_{d \in \mathcal{D}} v(\mathrm{rank}(d|r)) \lambda(\mathrm{rel}(d|u, q)) \\ &= \sum_{r} R(r) \sum_{d \in \mathcal{D}} v(\mathrm{rank}(d|r))\, u(d|q). \end{aligned}$$

While distributions R over rankings are still exponential in size, we can make use of the additional insight that utility can already be computed from the marginal rank distributions of the documents. Let $\mathbf{P}_{i,j}$ be the probability that R places document d_i at rank j, then $\mathbf{P}$ forms a doubly stochastic matrix of size $N \times N$, which means that the sum of each row and each column of the matrix is equal to 1. In other words, the sum of probabilities for each position is 1 and the sum of probabilities for each document is 1, i.e., $\sum_i \mathbf{P}_{i,j} = 1$ and $\sum_j \mathbf{P}_{i,j} = 1$. With knowledge of the doubly stochastic matrix $\mathbf{P}$, expected utility for a probabilistic ranking can be computed as

$$\mathrm{U}(\mathbf{P}|q) = \sum_{d_i \in \mathcal{D}} \sum_{j=1}^{N} \mathbf{P}_{i,j}\, u(d_i|q)\, v(j). \tag{3.1}$$

To make notation more concise, we can rewrite the utility of the ranking as a matrix product. For this, we introduce two vectors: $\mathbf{u}$ is a column vector of size N with $\mathbf{u}_i = u(d_i|q)$, and $\mathbf{v}$ is another column vector of size N with $\mathbf{v}_j = v(j)$. So, the expected utility (e.g., DCG) can be written as:

$$\mathrm{U}(\mathbf{P}|q) = \mathbf{u}^T \mathbf{P} \mathbf{v} \tag{3.2}$$

	Notation
Query	q
Relevance	rel
Utility	U
Deterministic Ranking	r
Position of item d in ranking r	rank($d|r$)
Probabilistic Ranking	R
Marginal Rank Distribution	$\mathbf{P}$
Probability of placing document d_i at rank j	$\mathbf{P}_{i,j}$
Position bias at position k	$v(k)$

Table 3.1: Notation for the ranking setup.

3.1.3 Optimizing Fair Rankings via Linear Programming

We will see in Section § 3.1.4 that not only does R imply a doubly stochastic matrix $\mathbf{P}$, but that we can also efficiently compute a probabilistic ranking R for every doubly stochastic matrix $\mathbf{P}$. We can, therefore, formulate the problem of finding the utility-maximizing probabilistic ranking under fairness constraints in terms of doubly stochastic matrices instead of distributions over rankings.

$$\begin{aligned}
\mathbf{P} = \mathrm{argmax}_{\mathbf{P}} \;\; & \mathbf{u}^T\mathbf{P}\mathbf{v} && \text{(expected utility)} \\
\text{s.t.} \;\; & \mathbb{1}^T\mathbf{P} = \mathbb{1}^T && \text{(sum of probabilities for each position)} \\
& \mathbf{P}\mathbb{1} = \mathbb{1} && \text{(sum of probabilities for each document)} \\
& 0 \le \mathbf{P}_{i,j} \le 1 && \text{(valid probability)} \\
& \mathbf{P} \text{ is fair} && \text{(fairness constraints)}
\end{aligned}$$

Note that the optimization objective is linear in N^2 variables $\mathbf{P}_{i,j}, 1 \le i, j \le N$. Furthermore, the constraints ensuring that $\mathbf{P}$ is doubly stochastic are linear as well, where $\mathbb{1}$ is the column vector of size N containing all ones. In the absence of a fairness constraint and for any $\mathbf{v}_j$ that decreases with j, the solution is the permutation matrix that ranks the set of documents in decreasing order of utility (conforming to the PRP).

Now that we have expressed the problem of finding the utility-maximizing probabilistic ranking, besides the fairness constraint, as a linear program, a convenient language to express fairness constraints would be linear constraints of the form

$$\mathbf{f}^T\mathbf{P}\mathbf{g} = h.$$

One or more of such constraints can be added, and the resulting linear program can still be solved efficiently and optimally with standard algorithms like the interior point method. As we will show in Section § 3.2, the vectors $\mathbf{f}$, $\mathbf{g}$ and the scalar h can be chosen to implement a range of different fairness constraints. To give some intuition, the vector $\mathbf{f}$ can be used to encode group identity and/or relevance of each document, while $\mathbf{g}$ will typically reflect the importance of each position (e.g., position bias).

3.1.4 Sampling Rankings

The solution **P** of the linear program is a matrix containing probabilities of each document at each position. To implement this solution in a ranking system, we need to compute a probabilistic ranking R that corresponds to **P**. From this probabilistic ranking, we can then sample rankings $r \sim R$ to present to the user[3]. Given the derivation of our approach, it is immediately apparent that the rankings r sampled from R fulfill the specified fairness constraints in expectation.

Computing R from **P** can be achieved via the Birkhoff-von Neumann (BvN) decomposition (Birkhoff, 1940), which provides a transformation to decompose a doubly stochastic matrix into a convex sum of permutation matrices. In particular, if **A** is a doubly stochastic matrix, there exists a decomposition of the form

$$\mathbf{A} = \theta_1\mathbf{A_1} + \theta_2\mathbf{A_2} + \cdots + \theta_n\mathbf{A_n}$$

where $0 \leq \theta_i \leq 1$, $\sum_i \theta_i = 1$, and where the $\mathbf{A_i}$ are permutation matrices (Birkhoff, 1940). In our case, the permutation matrices correspond to deterministic rankings of the document set and the coefficients correspond to the probability of sampling each ranking. According to the Marcus-Ree theorem, there exists a decomposition with no more than $(N-1)^2 + 1$ permutation matrices (Marcus and Ree, 1959). Such a decomposition can be computed efficiently in polynomial time using several algorithms (Chang et al., 1999; Dufossé and Uçar, 2016)[4].

[3] For usability reasons, it is preferable to make this sampling pseudo-random based on a hash of the user's identity, so that the same user receives the same ranking r if the same query is repeated. [4] For the experiments in this work, we use the implementation provided at `https://github.com/jfinkels/birkhoff`

3.1.5 Summary of the Algorithm

The following summarizes the algorithm for optimal ranking under fairness constraints. Note that we have assumed knowledge of the true relevances $u(d|q)$ throughout this work, whereas in practice one would work with estimates $\hat{u}(d|q)$ from some predictive model.

1. Set up the utility vector $\mathbf{u}$, the position discount vector $\mathbf{v}$, as well as the vectors $\mathbf{f}$ and $\mathbf{g}$, and the scalar h for the fairness constraints (see Section § 3.2).
2. Solve the linear program from Section § 3.1.3 for $\mathbf{P}$.
3. Compute the Birkhoff-von Neumann decomposition $\mathbf{P} = \theta_1\mathbf{P}_1 + \theta_2\mathbf{P}_2 + \cdots + \theta_n\mathbf{P}_n$.
4. Sample permutation matrix $\mathbf{P}_i$ with probability proportional to θ_i and display the corresponding ranking r_i.

Note that the rankings sampled in the last step of the algorithm fulfill the fairness constraints in expectation, while at the same time they maximize expected utility.

3.2 Constructing Group Fairness Constraints

Now that we have established a framework for formulating fairness constraints and optimally solving the associated ranking problem, we still need to understand the expressiveness of constraints of the form $\mathbf{f}^T\mathbf{P}\mathbf{g} = h$. In this section, we explore how three concepts from algorithmic fairness – demographic parity, disparate exposure, and disparate impact – can be implemented in our framework and thus be enforced efficiently and with provable guarantees. All these constraints aim to allocate exposure fairly and can be formally defined in our framework as follows.

Position Bias and Exposure. Let $\mathbf{v}_j$ represent the importance of position j, or more concretely the position bias at j, which is the fraction of users that examine the item at this position. Then we define exposure for a document d_i under a probabilistic ranking $\mathbf{P}$ as

$$\text{Exposure}(d_i|\mathbf{P}) = \sum_{j=1}^{N} \mathbf{P}_{i,j}\mathbf{v}_j. \tag{3.3}$$

The goal is to allocate exposure fairly between groups G_k. Documents and items may belong to different groups because of some sensitive attributes – for example, news stories belong to different sources, products belong to different manufacturers, or applicants grouped by gender. The fairness constraints, that we will formulate in the following, implement different goals for allocating exposure between groups.

To illustrate the effect of the fairness constraints, we will provide empirical results on two ranking problems. For both, we use the average relevance of each document (normalized between 0 and 1) as the utility $\mathbf{u}_i = u(d_i|q)$ and set the position bias to $\mathbf{v}_j = \frac{1}{\log(1+j)}$ just like in the standard definition of DCG. More generally, one can also plug in the actual position-bias value, which can be estimated through an intervention experiment (Joachims et al., 2017), or other observational methods (Agarwal et al., 2019b).

Job-seeker example. Let us consider the job-seeker example illustrated in Figure 3.1. The ranking problem consists of 6 applicants with probabilities of relevance to an employer of $\mathbf{u} = (0.82, 0.81, 0.80, 0.79, 0.78, 0.77)^T$. Groups G_0 and G_1 reflect gender, with the first three applicants belonging to the male group and the last three to the female group. Note that this example is similar to the example presented in Section § 1.1.1 as it raises the same concern about fair allocation of exposure (as illustrated in the figure).

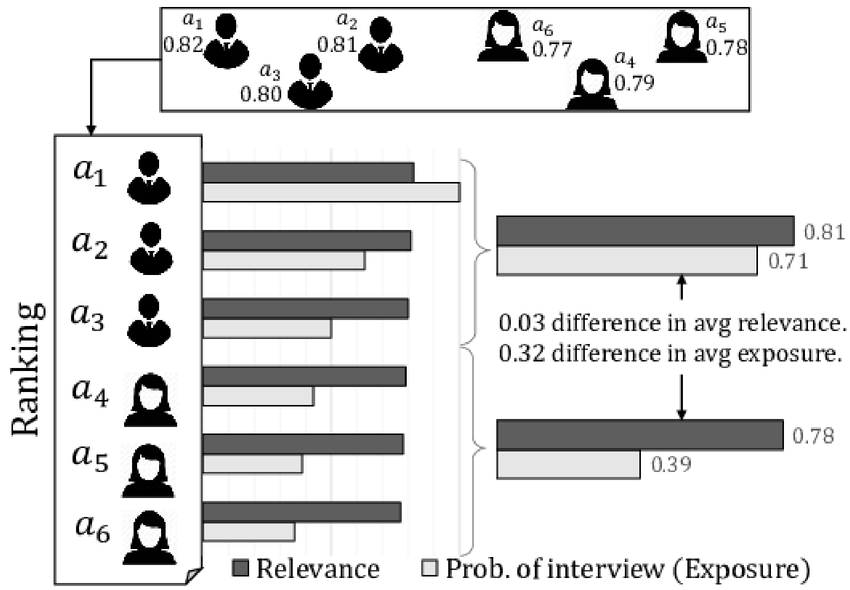


Figure 3.1: Job seeker example: The example illustrates how small a difference in relevance can lead to a large difference in exposure (an opportunity) between the group of male and female candidates.

News recommendation dataset. We use a subset of the *Yow* news recommendation dataset (Zhang, 2005) to analyze our method on a larger and real-world relevance distribution. The dataset contains explicit and implicit feedback from a set of users for news articles from different RSS feeds. We randomly sample a subset of news articles in the "people" topic coming from the top two sources. The sources are identified using their RSS Feed identifier and used as groups G_0 and G_1. The *relevant* field in the dataset is used as the measure of relevance for our task. Since the relevance is given as a rating from 1 to 5, we divide it by 5 and add a small amount of Gaussian noise ($\mu = 0, \epsilon = 0.05$) to break ties. The resulting $\mathbf{u}_i$ are clipped to lie between 0 and 1.

In the following, we formulate fairness constraints using three ideas for allocation of exposure to different groups. In particular, we will define constraints of the form $\mathbf{f}^T\mathbf{P}\mathbf{g} = h$ for the optimization problem in § 3.1.3. For simplicity, we will only present the case of a binary-valued sensitive attribute, i.e., two groups G_0 and G_1. However, these constraints may be defined for each pair of groups and for each sensitive attribute, and be included in the linear program.

3.2.1 Demographic Parity Constraints

Arguably the simplest way of defining fairness of exposure between groups is to enforce that the average exposure of the documents in both the groups is equal. Denoting average exposure in a group with

$$\text{Exposure}(G_k|\mathbf{P}) = \frac{1}{|G_k|}\sum_{d_i \in G_k} \text{Exposure}(d_i|\mathbf{P}),$$

this can be expressed as the following constraint in our framework:

$$\begin{aligned} & \text{Exposure}(G_0|\mathbf{P}) = \text{Exposure}(G_1|\mathbf{P}) && (3.4)\\ \Leftrightarrow\ & \frac{1}{|G_0|}\sum_{d_i \in G_0}\sum_{j=1}^{N}\mathbf{P}_{i,j}\mathbf{v}_j = \frac{1}{|G_1|}\sum_{d_i \in G_1}\sum_{j=1}^{N}\mathbf{P}_{i,j}\mathbf{v}_j && (3.5)\\ \Leftrightarrow\ & \sum_{d_i \in \mathcal{D}}\sum_{j=1}^{N}\left(\frac{\mathbb{1}_{d_i \in G_0}}{|G_0|} - \frac{\mathbb{1}_{d_i \in G_1}}{|G_1|}\right)\mathbf{P}_{i,j}\mathbf{v}_j = 0 && (3.6)\\ \Leftrightarrow\ & \mathbf{f}^T P\mathbf{v} = 0. && \left(\text{with } \mathbf{f}_i = \tfrac{\mathbb{1}_{d_i \in G_0}}{|G_0|} - \tfrac{\mathbb{1}_{d_i \in G_1}}{|G_1|}\right) \end{aligned}$$

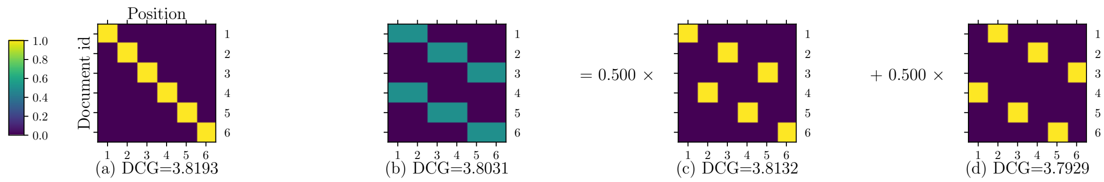


Figure 3.2: Job seeker example with demographic parity constraint. (a) Optimal unfair ranking that maximizes DCG. (b) Optimal fair ranking under demographic parity. (c) and (d) are the BvN decomposition of the fair ranking.

In the last step, we obtain a constraint in the form $\mathbf{f}^T\mathbf{P}\mathbf{g} = h$ which one can plug it into the linear program from Section §3.1.3. We call this a *Demographic Parity Constraint* similar to an analogous constraint in fair supervised learning (Calders et al., 2009; Zliobaite, 2015). Similar to that setting, in our case as well, such a constraint may lead to a big loss in utility in cases when the two groups are very different in terms of relevance distribution.

Experiments. We solved the linear program in Section §3.1.3 twice – once without and once with the demographic parity constraint from above. For the job seeker example, Figure 3.2 shows the optimal rankings in terms of $\mathbf{P}$ without and with fairness constraint in panels (a) and (b), respectively. The color indicates the probability value.

Note that the fair ranking according to demographic parity includes a substantial amount of stochasticity. However, panels (c) and (d) show that the fair ranking can be decomposed into a mixture of two deterministic permutation matrices with the associated weights.

Compared to the DCG of the unfair ranking with 3.8193, the optimal fair ranking has a slightly lower utility with a DCG of 3.8031. However, the drop in utility due to the demographic parity constraint could be substantially larger. For example, if we lowered the relevances for the group G_1 to $\mathbf{u} = (0.82, 0.81, 0.80, 0.03, 0.02, 0.01)^T$, we would still get the same fair ranking as the current solution, since this fairness constraint is ignorant of relevance. In this ranking, roughly every second document has low relevance, leading to a large drop in DCG. It is interesting to point out that the effect of demographic parity in ranking is therefore analogous to its effect in supervised learning, where it can also lead to a large drop in classification accuracy (Dwork et al., 2012).

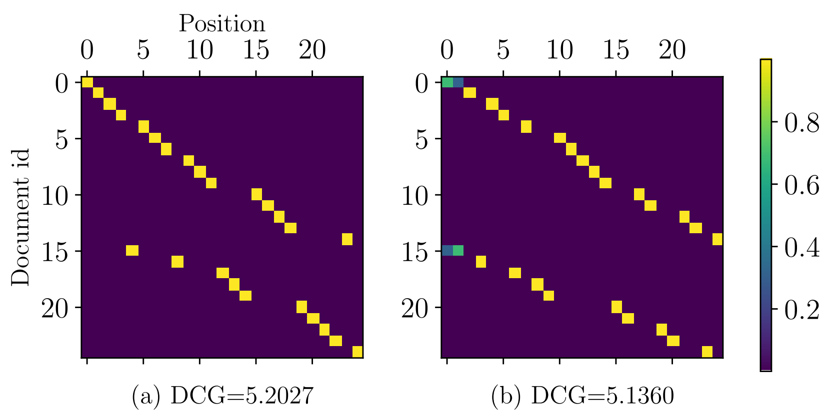


Figure 3.3: News recommendation dataset with demographic parity constraint. G_0: Document ID. 0-14, G_1: 15-24 (a) Optimal unfair ranking that maximizes DCG. (b) Optimal fair ranking under demographic parity.

We also conducted the same experiment on the news recommendation dataset. Figure 3.3 shows the optimal ranking matrix and the fair probabilistic ranking along with DCG for each. Note that even though the optimal unfair ranking places documents from G_1 starting at position 5, the constraint pushes the ranking of the news items from G_1 further up the ranking starting either at rank 1 or rank 2. In this case, the optimal fair ranking happens to be (almost) deterministic except at the beginning.

3.2.2 Disparate Exposure Constraints

Unlike demographic parity, the constraints we explore in this and the following section depend on the relevance of the items being ranked. In this way, these constraints have the potential to address the concerns for the job-seeker example, where a small difference in relevance was magnified into a large difference in exposure. Denoting the average utility of a group with

$$\mathrm{U}(G_k|q) = \frac{1}{|G_k|} \sum_{d_i \in G_k} \mathbf{u}_i,$$

this motivates the following type of constraint, which enforces that exposure of the two groups to be proportional to their average utility.

$$\begin{aligned}
& \frac{\mathrm{Exposure}(G_0|\mathbf{P})}{\mathrm{U}(G_0|q)} = \frac{\mathrm{Exposure}(G_1|\mathbf{P})}{\mathrm{U}(G_1|q)} \\
\Leftrightarrow\ & \frac{\frac{1}{|G_0|} \sum_{d_i \in G_0} \sum_{j=1}^{N} \mathbf{P}_{i,j} \mathbf{v}_j}{\mathrm{U}(G_0|q)} = \frac{\frac{1}{|G_1|} \sum_{d_i \in G_1} \sum_{j=1}^{N} \mathbf{P}_{i,j} \mathbf{v}_j}{\mathrm{U}(G_1|q)} && (3.7) \\
\Leftrightarrow\ & \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\frac{\mathbb{1}_{d_i \in G_0}}{|G_0| \mathrm{U}(G_0|q)} - \frac{\mathbb{1}_{d_i \in G_1}}{|G_1| \mathrm{U}(G_1|q)} \right) \mathbf{P}_{i,j} \mathbf{v}_j = 0 && (3.8) \\
\Leftrightarrow\ & \mathbf{f}^T P \mathbf{v} = 0. && \left(\text{with } \mathbf{f}_i = \left(\frac{\mathbb{1}_{d_i \in G_0}}{|G_0| \mathrm{U}(G_0|q)} - \frac{\mathbb{1}_{d_i \in G_1}}{|G_1| \mathrm{U}(G_1|q)} \right)\right)
\end{aligned}$$

In (Singh and Joachims, 2018), we name this constraint as the *Disparate Treatment* constraint because giving exposure to a group is analogous to treating the group of documents favorably. In principle, this is motivated by the "Recommendations as Treatments" concept from Schnabel et al. (2016), where recommending or exposing a document to the user is considered analogous to treatment as in causal inference methods, and the user's click or purchase is considered the effect or the impact of the treatment. However, note that this constraint may not translate to the legally recognized types of illegal discrimination called Disparate Treatment and Disparate Impact for all applications.

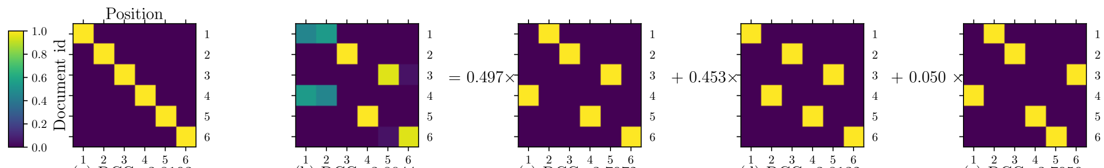


Figure 3.4: Job seeker example with disparate exposure constraint. (a) Optimal unfair ranking. (b) Fair ranking under disparate exposure constraint. (c), (d), (e) are the BvN decomposition of the fair ranking.

To quantify exposure disparity, we also define a measure called Disparate Treatment Ratio (DTR) to evaluate how unfair a ranking is in this respect, i.e., how differently the two groups are treated.

$$\mathrm{DTR}(G_0, G_1|\mathbf{P}, q) = \frac{\mathrm{Exposure}(G_0|\mathbf{P})/\mathrm{U}(G_0|q)}{\mathrm{Exposure}(G_1|\mathbf{P})/\mathrm{U}(G_1|q)}$$

Note that this ratio equals one if the disparate exposure constraint in Equation 3.7 is fulfilled. Whether the value is less than 1 or greater than 1 tells which group out of G_0 or G_1 is disadvantaged in terms of disparate exposure.

Experiments. We again compute the optimal ranking without fairness constraint and with the disparate exposure constraint. The results for the job-seeker example are shown in Figure 3.4. The figure also shows the BvN decomposition of the resultant probabilistic ranking into three permutation matrices. As expected, the fair ranking has an optimal DTR while the unfair ranking has a DTR of 1.7483. Also expected is that the fair ranking has a lower DCG than the optimal deterministic ranking, but that it has a higher DCG than the optimal fair ranking under demographic parity.

We conducted the same experiment for the news recommendation dataset. Figure 3.5 shows the optimal ranking matrix and the fair probabilistic ranking along with DCG for each. Here, the ranking computed without the fairness

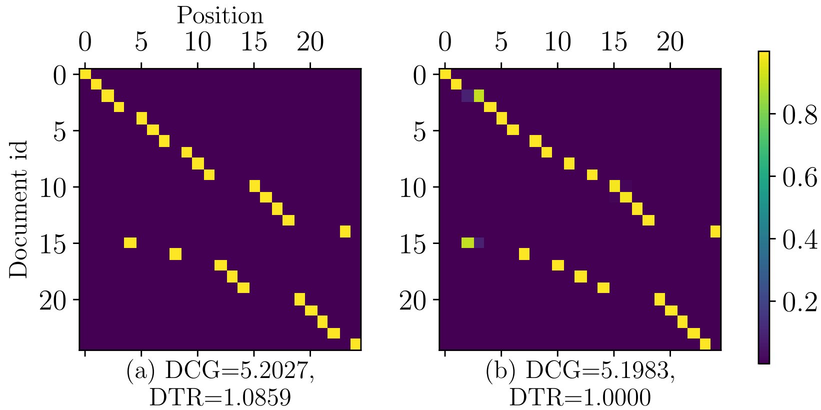


Figure 3.5: News recommendation dataset with disparate exposure constraint. (a) Optimal unfair ranking. (b) Fair ranking under disparate exposure constraint.

constraint happened to be almost fair according to disparate exposure already, and the fairness constraint has very little impact on DCG.

3.2.3 Disparate Impact Constraints

In the previous section, we constrained the exposures for the two groups to be proportional to their average utility. However, we may want to go a step further and define a constraint on the impact, e.g., the expected clickthrough or purchase rate as these more directly reflect the economic impact of the ranking. In particular, we may want to assure that the clickthrough rates for the groups as determined by the exposure and relevance are proportional to their average utility. To formally define this, let us first model the probability of a document getting clicked according to the following simple click model (Joachims et al., 2005; Craswell et al., 2008; Richardson et al., 2007):

$$\begin{aligned} P(\text{click on document } i) &= P(\text{examining } i) \times P(i \text{ is relevant}) \\ &= \text{Exposure}(d_i|\mathbf{P}) \times P(i \text{ is relevant}) \end{aligned}$$

$$= \left(\sum_{j=1}^{N} \mathbf{P}_{i,j} \mathbf{v}_j \right) \times \mathbf{u}_i.$$

We can now compute the average clickthrough rate of documents in a group G_k as

$$\mathrm{CTR}(G_k|\mathbf{P}) = \frac{1}{|G_k|} \sum_{i \in G_k} \sum_{j=1}^{N} \mathbf{P}_{i,j} \mathbf{u}_i \mathbf{v}_j.$$

The following *Disparate Impact Constraint* enforces that the expected clickthrough rate of each group is proportional to its average utility:

$$\frac{\mathrm{CTR}(G_0|\mathbf{P})}{\mathrm{U}(G_0|q)} = \frac{\mathrm{CTR}(G_1|\mathbf{P})}{\mathrm{U}(G_1|q)} \tag{3.9}$$

$$\Leftrightarrow \frac{\frac{1}{|G_0|} \sum_{i \in G_0} \sum_{j=1}^{N} \mathbf{P}_{i,j} \mathbf{u}_i \mathbf{v}_j}{\mathrm{U}(G_0|q)} = \frac{\frac{1}{|G_1|} \sum_{i \in G_1} \sum_{j=1}^{N} \mathbf{P}_{i,j} \mathbf{u}_i \mathbf{v}_j}{\mathrm{U}(G_1|q)} \tag{3.10}$$

$$\Leftrightarrow \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\frac{\mathbb{1}_{d_i \in G_0}}{|G_0| \mathrm{U}(G_0|q)} - \frac{\mathbb{1}_{d_i \in G_1}}{|G_1| \mathrm{U}(G_1|q)} \right) \mathbf{u}_i \mathbf{P}_{i,j} \mathbf{v}_j = 0 \tag{3.11}$$

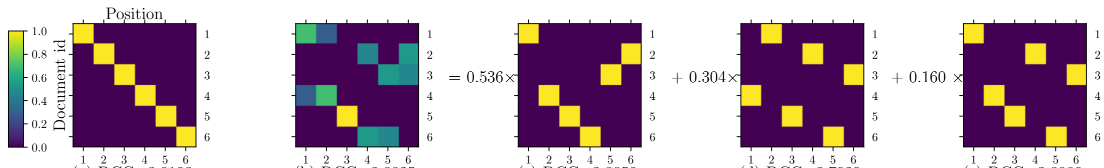
$$\Leftrightarrow \mathbf{f}^T P \mathbf{v} = 0. \qquad \left(\text{with } \mathbf{f}_i = \left(\frac{\mathbb{1}_{d_i \in G_0}}{|G_0| \mathrm{U}(G_0|q)} - \frac{\mathbb{1}_{d_i \in G_1}}{|G_1| \mathrm{U}(G_1|q)} \right) \mathbf{u}_i \right)$$

Similar to DTR, we can define the following Disparate Impact Ratio (DIR) to measure the extent to which the disparate impact constraint is violated:

$$\mathrm{DIR}(G_0, G_1|\mathbf{P}, q) = \frac{\mathrm{CTR}(G_0|\mathbf{P})/\mathrm{U}(G_0|q)}{\mathrm{CTR}(G_1|\mathbf{P})/\mathrm{U}(G_1|q)}$$

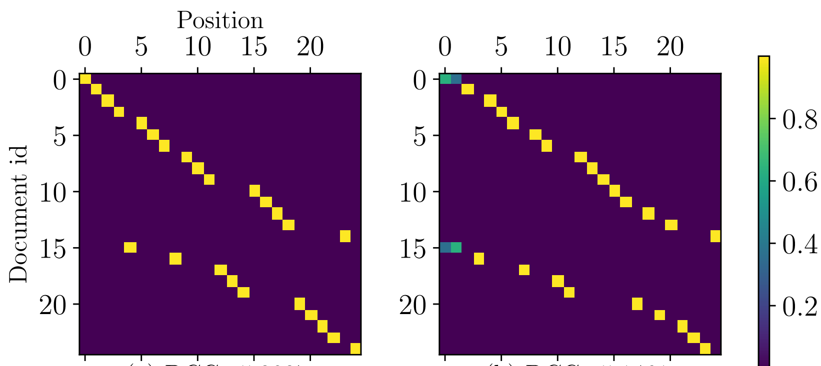
Note that this ratio equals one if the disparate impact constraint in Equation 3.11 is fulfilled. Similar to DTR, whether DIR is less than 1 or greater than 1 tells which group is disadvantaged in terms of disparate impact.

Experiments. We again compare the optimal rankings with and without the fairness constraint. The results for the job-seeker example are shown in Figure 3.6. Again, the optimal fair ranking has a BvN decomposition into three deterministic rankings, and it has a slightly reduced DCG. However, there is a



(a) DCG=3.8193, DIR=1.8193
(b) DCG=3.8025, DIR=1.0000
(c) DCG=3.8059, DIR=1.1192
(d) DCG=3.7929, DIR=0.7501
(e) DCG=3.8089, DIR=1.1801

Figure 3.6: Job seeker example with disparate impact constraint. (a) Optimal unfair ranking. (b) Fair ranking under disparate impact constraint. (c), (d), (e) are the BvN decomposition of the fair ranking.



(a) DCG=5.2027, DIR=1.5211
(b) DCG=5.1461, DIR=1.0000

Figure 3.7: News recommendation dataset with disparate impact constraint. (a) Optimal unfair ranking. (b) Fair ranking under disparate impact constraint.

large improvement in DIR from the fairness constraint since the PRP ranking has a substantial disparate impact on the two groups.

The results for the news recommendation dataset are given in Figure 3.7, where we also see a large improvement in DIR. The DCG is lower than the unconstrained DCG and the DCG with disparate exposure constraint but higher than the DCG with demographic parity constraint.

3.2.4 Discussion

In the last section, we implemented three fairness constraints in our framework, motivated by the concepts of demographic parity, disparate exposure, and disparate impact. The primary purpose was to explore the expressiveness of the framework, and we do not argue that these constraints are the only conceivable ones or the correct ones for a given application. In particular, it appears that fairness in rankings is inherently a trade-off between the utility of the users and the rights of the items that are being ranked and that different applications require making this trade-off in different ways. For example, we may not want to convey strong rights to the books in a library when a user is trying to locate a book, but the situation is different when candidates are being ranked for a job opening. We, therefore, focused on creating a flexible framework that covers a substantial range of fairness constraints.

3.2.4.1 Relationship with related work

Concurrent and independent work by Biega et al. (2018) formulates fairness for rankings similar to a special case of our framework, aiming to achieve amortized fairness of attention by making exposure proportional to relevance. They focus on individual fairness, which in our framework amounts to the special case of protected groups of size one. The two approaches not only differ in expressive power, but algorithmically, they solve an integer linear program to generate a series of rankings, while our approach provides a provably efficient solution via a standard linear program and the Birkhoff-von Neumann decomposition (Birkhoff, 1940).

Most of the composition-based fairness constraints for ranking defined in the previous work reflect parity constraints restricting the fraction of items with each attribute in the ranking as mentioned in Section § 2.3.2 (e.g., (Yang and Stoyanovich, 2017; Celis et al., 2018; Zehlike et al., 2017; Mehrotra et al., 2018)). In comparison, we proposed a general algorithmic framework for efficiently computing optimal probabilistic rankings for a large class of possible fairness constraints. Although we primarily focus on three fairness constraints, with only a few minor modifications, these parity constraints from related works can also be instantiated in the framework proposed.

3.2.4.2 Group fairness vs. individual fairness

In our experiments, we observe that even though the constraints ensure that the rankings have no disparate exposure or disparate impact across groups, individual items within a group might still be considered to suffer from disparate exposure or impact. For example, in the job-seeker experiment for disparate exposure (Figure 3.4), the allocation of exposure to the candidates within the group G_0 still follows the same exposure drop-off going down the ranking that we considered unfair according to the disparate exposure constraint. As a remedy, one could include additional fairness constraints for other sensitive attributes, like race, disability, and national origin to further refine the desired notion of fairness. In the extreme, our framework allows protected groups of size one, such that it can also express notions of individual fairness. For example, in the case of disparate exposure, we could express individual fairness as a set of $N - 1$ constraints over N groups of size one, resulting in a notion of fairness similar to Biega et al. (2018). However, for the Disparate Impact constraint where the expected clickthrough rates are proportional to the relevances, it is unclear whether individual fairness makes sense unless we rank items uniformly at random.

3.2.4.3 Using estimated utilities

In our definitions and experiments, we assumed that we have access to the true expected utilities $u(d|q)$ (defined as a function of relevance). In practice, these utilities are typically estimated via machine learning. This learning step is subject to other biases that may, in turn, lead to biased estimates $\hat{u}(d|q)$. Most importantly, biased estimates may be the result of selection biases in click data, but recent counterfactual learning techniques (Joachims et al., 2017) have been shown to permit unbiased learning-to-rank despite biased click data. Chapters 4 and 5 deal with the case when utilities need to be estimated as a part of the problem.

3.2.4.4 Cost of fairness

Including the fairness constraints in the optimization problem comes at the cost of effectiveness as measured by DCG and other conventional measures. This loss in utility can be computed as $CoF = \mathbf{u}^T(\mathbf{P}^* - \mathbf{P})\mathbf{v}$, where $\mathbf{P}^*$ is the deterministic optimal ranking, and $\mathbf{P}$ represents the fair ranking. We have already discussed that this cost can be substantial for the demographic parity constraint. In particular, for demographic parity, it is easy to see that the utility of the fair ranking approaches zero if all relevant documents are in one group, and the size of the other group approaches infinity.

3.2.4.5 Feasibility of fair solutions

The linear program may not have a solution in extreme conditions, corresponding to cases where no fair solution exists with respect to the chosen constraint(s). Consider the disparate exposure constraint

$$\frac{\text{Exposure}(G_0|\mathbf{P})}{\text{Exposure}(G_1|\mathbf{P})} = \frac{\text{U}(G_0|q)}{\text{U}(G_1|q)}.$$

We can adversarially construct an infeasible constraint by choosing the relevance so that the ratio on the RHS lies outside the range that LHS can achieve by varying $\mathbf{P}$. The maximum of the LHS occurs when all the documents of G_0 are placed above all the documents of G_1, and vice versa for the minimum.

$$\max\left\{\frac{\text{Exposure}(G_0|\mathbf{P})}{\text{Exposure}(G_1|\mathbf{P})}\right\} = \frac{\sum_{j=1}^{|G_0|}\mathbf{v}_j}{\sum_{j=|G_0|+1}^{|G_0|+|G_1|}\mathbf{v}_j}, \quad \text{(all } G_0 \text{ documents in top } |G_0| \text{ positions)}$$

$$\min\left\{\frac{\text{Exposure}(G_0|\mathbf{P})}{\text{Exposure}(G_1|\mathbf{P})}\right\} = \frac{\sum_{j=|G_1|+1}^{|G_1|+|G_0|}\mathbf{v}_j}{\sum_{j=1}^{|G_1|}\mathbf{v}_j}$$

$$\text{(all } G_0 \text{ documents in bottom } |G_0| \text{ positions)}$$

Hence, a fair ranking according to disparate exposure only exists if the ratio of average utilities lies within the range of possible values for the exposure:

$$\frac{\sum_{j=|G_1|+1}^{|G_1|+|G_0|}\mathbf{v}_j}{\sum_{j=1}^{|G_1|}\mathbf{v}_j} \leq \frac{\text{U}(G_0|q)}{\text{U}(G_1|q)} \leq \frac{\sum_{j=1}^{|G_0|}\mathbf{v}_j}{\sum_{j=|G_0|+1}^{|G_0|+|G_1|}\mathbf{v}_j}$$

However, in such a scenario, the constraint can still be satisfied if we introduce more documents belonging to neither group (or the group with more relevant documents). This increases the range of the LHS, and the ranking doesn't have to give undue exposure to one of the groups.

It is worth noting that the feasibility criteria derived here are merely a property of the fairness constraints defined in this chapter and might not correspond to an actual fairness concern. This is simply because some documents may receive *extra* exposure that does not come at the expense of other documents, e.g., the case where only one group of items is relevant to the query. We discuss this issue in detail in the next chapter in Section § 4.1.2 and propose an alternate formulation that alleviates this issue.

3.3 Summary

In this chapter, we considered the fairness of rankings through the lens of exposure allocation between groups. Instead of defining a single notion of fairness, we developed a general framework that employs probabilistic rankings and linear programming to compute the utility-maximizing ranking under a whole class of fairness constraints. To verify the expressiveness of this class, we showed how to express fairness constraints motivated by the concepts of demographic parity, disparate exposure, and disparate impact. We conjecture that the appropriate definition of fair exposure depends on the application, which makes this expressiveness desirable.

CHAPTER 4

LEARNING-TO-RANK WITH FAIRNESS CONSTRAINTS

In the previous chapter, we defined fairness of ranking corresponding to how rankings allocate exposure to individual items or groups of items based on their merit, where merit is defined as a function of the relevance of the item to the user. We specify and enforce fairness constraints that explicitly link merit to exposure in expectation or amortized over a set of queries. However, we assumed that the relevances of all items are either already known or estimated, and did not address the problem of learning-to-rank (LTR) (Section § 2.2.3) where the ranking system estimates the relevances or their relative ordering to learn a ranking function.

A straightforward way of applying the approach from Chapter 3 would be to estimate the relevance using a regression method and then use the linear program as a post-processing step to obtain a fair ranking distribution. However, there are at least two pitfalls of this approach — first, a fairness-agnostic estimation of relevances from query-item features might not be able to detect and ignore biased features, and second, a biased estimate of relevances leads to the enforcement of the wrong fairness constraints since the Disparate Exposure and Impact constraints are defined with respect to these estimates. In the worst-case scenario, these two drawbacks may even lead to the opposite effect of what a fair ranking intends to achieve. In this chapter[1], we propose an end-to-end LTR approach that considers fairness as one of the objectives while training and hence can avoid these drawbacks.

[1] This chapter is based on joint work with Thorsten Joachims (Singh and Joachims, 2019)

In this chapter, we develop FAIR-PG-RANK which is a Learning-to-Rank (LTR) algorithm that maximizes utility to the users as well as rigorously enforces merit-based exposure constraints towards the items. Focusing on the notions of fairness that ensure that the relative allocation of exposure is based on the items' merit and by considering the constraints already during learning, we find that FAIR-PG-RANK can identify biases in the representation that post-processing methods (Singh and Joachims, 2018; Biega et al., 2018) are, by design, unable to detect. Furthermore, we find that FAIR-PG-RANK performs better than other heuristic approaches from the literature (Zehlike and Castillo, 2020).

From a technical perspective, the main contributions of this work are three-fold. First, we develop a conceptual framework in which it is possible to formulate fair LTR as a policy-learning problem subject to fairness constraints. We show that viewing fair LTR as learning a stochastic ranking policy leads to a rigorous formulation that can be addressed via Empirical Risk Minimization (ERM) on both the utility and the fairness constraint. Second, we propose a class of fairness constraints for ranking that incorporates notions of both individual and group fairness. And, third, we propose a policy-gradient method for implementing the ERM procedure that can directly optimize any information retrieval utility metric and a wide range of fairness criteria. Across a number of empirical evaluations, we find that the policy-gradient approach is a competitive LTR method in its own right, that FAIR-PG-RANK can identify and avoid biased features when trading-off utility for fairness, and that it can effectively optimize notions of individual and group fairness on real-world datasets.

4.1 Policy Learning for Ranking

The key goal of this work is to learn ranking policies where the allocation of exposure to items is not an accidental by-product of maximizing utility to the users, but where one can specify a merit-based exposure-allocation constraint that is enforced by the learning algorithm. An illustrative example adapted from Section § 1.1.1 is that of ranking 10 job candidates, where the estimated probabilities of relevance (e.g., probability that an employer will invite for an interview) of 5 male job candidates are $\{0.89, 0.89, 0.89, 0.89, 0.89\}$ and those of 5 female candidates are $\{0.88, 0.88, 0.88, 0.88, 0.88\}$. Notice that these probabilities of relevance can themselves be gender-biased because of biased data or a biased prediction model. If these 10 candidates were ranked by these probabilities of relevance – thus maximizing utility to the users under virtually all information retrieval metrics (Robertson, 1977) – the female candidates would get far less exposure (ranked 6,7,8,9,10) than the male candidates (ranked 1,2,3,4,5) even though they have almost the same estimated relevance. In this way, the ranking function itself is responsible for creating a strong *endogenous* bias against the female candidates, greatly amplifying and thus perpetuating any *exogenous* bias that may have led to small differences in the relevance estimates.

Addressing the endogenous bias created by the system itself, we argue that it should be possible to explicitly specify how exposure is allocated (e.g. make exposure proportional to relevance), that this specified exposure allocation is truthfully learned by the ranking policy (e.g. no systematic bias towards one of the groups), and that the ranking policy maintains a high utility to the users.

Generalizing from this illustrative example, we develop our fair LTR framework as guided by the following three goals:

Goal 1: Exposure allocated to an item is based on its merit. More merit means more exposure.

Goal 2: Enable the explicit statement of how exposure is allocated relative to the merit of the items.

Goal 3: Optimize the utility of the rankings to the users while satisfying *Goal 1* and *Goal 2*.

We will illustrate and further refine these goals as we develop our framework in the rest of this section. In particular, we first formulate the LTR problem in the context of empirical risk minimization (ERM) where exposure-allocation constraints are included in the empirical risk. We then define concrete families of allocation constraints for both individual and group fairness.

4.1.1 Learning to Rank as Policy Learning via ERM

Let $\mathcal{Q}$ be the distribution from which queries are drawn. Each query q has a candidate set of documents $d^q = \{d_1^q, d_2^q, \dots d_{n(q)}^q\}$ that needs to be ranked, and a corresponding set of real-valued relevance judgments, $\mathrm{rel}^q = (\mathrm{rel}_1^q, \mathrm{rel}_2^q \dots \mathrm{rel}_{n(q)}^q)$. Our framework is agnostic to how relevance is defined, and it could be the probability that a user with query q finds the document relevant, or it could be some subjective judgment of relevance as assigned by a relevance judge. Finally, each document d_i^q is represented by a feature vector $x_i^q = \Psi(q, d_i^q)$ that describes the match between document d_i^q and query q.

We consider stochastic ranking functions $\pi \in \Pi$, where $\pi(r|q)$ is a distribution over the rankings r (i.e., permutations) of the candidate set. We refer to π as a ranking policy and note that deterministic ranking functions are merely a special case. However, a key advantage of considering the full space of stochastic ranking policies is their ability to distribute expected exposure in a continuous fashion, which provides more fine-grained control and enables gradient-based optimization.

The conventional goal in LTR is to find a ranking policy π^* that maximizes the expected utility of π

$$\pi^* = \text{argmax}_{\pi\in\Pi}\, \mathbb{E}_{q\sim\mathcal{Q}}[U(\pi|q)],$$

where the utility of a stochastic policy π for a query q is defined as the expectation of a ranking metric Δ over π

$$U(\pi|q) = \mathbb{E}_{r\sim\pi(r|q)}\left[\Delta(r, \text{rel}^q)\right].$$

Common choices for Δ are DCG, NDCG, Average Rank, or ERR. For concreteness, we focus on NDCG as in (Chapelle and Chang, 2011), which is the normalized version of $\Delta_{\text{DCG}}(r, \text{rel}^q) = \sum_{j=1}^{n_q} \frac{u(r(j)|q)}{\log(1+j)}$, where $u(r(j)|q)$ is the utility of the document placed by ranking r on position j for q as a function of relevance (e.g., $u(i|q) = 2^{\text{rel}_i^q} - 1$). NDCG normalizes DCG via $\Delta_{\text{NDCG}}(r, \text{rel}^q) = \frac{\Delta_{\text{DCG}}(r,\text{rel}^q)}{\max_r \Delta_{\text{DCG}}(r,\text{rel}^q)}$.

Fair Ranking policies. Instead of single-mindedly maximizing this utility measure like in conventional LTR algorithms, we include a constraint into the learning problem that enforces an application-dependent notion of fair allocation of exposure. To this effect, let's denote with $\mathcal{D}(\pi|q) \geq 0$ a measure of unfairness or the disparity, which we will define in detail in Section §4.1.2. We can now formulate the objective of fair LTR by constraining the space of admissible ranking policies to those that have expected disparity less than some parameter δ.

$$\pi^*_\delta = \text{argmax}_\pi\, \mathbb{E}_{q\sim Q}\left[U(\pi|q)\right] \;\; \text{s.t.}\;\; \mathbb{E}_{q\sim Q}\left[\mathcal{D}(\pi|q)\right] \leq \delta$$

Since we only observe samples from the query distribution Q, we resort to the Empirical Risk Minimization principle (ERM) and estimate the expectations with their empirical counterparts. Denoting the training set as $\mathcal{T} = \{(\mathbf{x}^q, \text{rel}^q)\}_{q=1}^N$, the empirical analog of the optimization problem becomes

$$\hat{\pi}^*_\delta = \text{argmax}_\pi\, \frac{1}{N}\sum_{q=1}^{N} U(\pi|q) \;\; \text{s.t.}\;\; \frac{1}{N}\sum_{q=1}^{N} \mathcal{D}(\pi|q) \leq \delta.$$

Using a Lagrange multiplier, this is equivalent to

$$\hat{\pi}^*_\delta = \text{argmax}_\pi \min_{\lambda\geq 0} \frac{1}{N}\sum_{q=1}^{N} U(\pi|q) - \lambda\left(\frac{1}{N}\sum_{q=1}^{N} \mathcal{D}(\pi|q) - \delta\right).$$

In the following, we avoid minimization w.r.t. λ for a chosen δ. Instead, we steer the utility/fairness trade-off by choosing a particular λ and then computing the corresponding δ afterwards. This means we merely have to solve

$$\hat{\pi}^*_\lambda = \text{argmax}_\pi\, \frac{1}{N}\sum_{q=1}^{N} U(\pi|q) - \lambda\frac{1}{N}\sum_{q=1}^{N} \mathcal{D}(\pi|q) \tag{4.1}$$

and then recover $\delta_\lambda = \frac{1}{N}\sum_{q=1}^{N} \mathcal{D}(\hat{\pi}^*_\lambda|q)$ afterwards. Note that this formulation implements our third goal from the opening paragraph, although we still have to give a concrete definition of $\mathcal{D}$.

4.1.2 Defining a Class of Fairness Measures for Rankings

To make the training objective in Equation (4.1) fully specified, we still need a concrete definition of the unfairness measure $\mathcal{D}$. To this effect, we adopt the fair exposure allocation framework from Chapter 3 since it allows a wide range of application-dependent notions of group-based fairness, including Statistical Parity, Disparate Exposure, and Disparate Impact. In order to formulate any specific disparity measure $\mathcal{D}$, we first need to define position bias and exposure.

Position Bias. The position bias of position j, $\mathbf{v}_j$, is defined as the fraction of users accessing a ranking who examine the item at position j. This captures how much attention an item will receive, where higher positions are expected to receive more attention than lower positions. In operational systems, position bias can be directly measured using eye-tracking (Joachims et al., 2007), or indirectly estimated through swap experiments (Joachims et al., 2017) or intervention harvesting (Agarwal et al., 2019b; Fang et al., 2019).

Exposure. For a given query q and ranking distribution $\pi(r|q)$, the exposure of a document is defined as the expected attention that a document receives. This is equivalent to the expected position bias from all the positions that the document can be placed in. Exposure is denoted as $v_\pi(d_i)$ and can be expressed as

$$\text{Exposure}(d_i|\pi) = v_\pi(d_i) = \mathbb{E}_{r\sim\pi(r|q)}\left[\mathbf{v}_{r(d_i)}\right], \tag{4.2}$$

where $r(d_i)$ is the position of document d_i under ranking r.

Allocating exposure based on merit. Our first two goals from the opening paragraph postulate that exposure should be based on an application-dependent notion of merit. We define the *merit* of a document as a function of its relevance to the query (e.g., rel_i, rel_i^2 or $\sqrt{\text{rel}_i}$ depending on the application). Let's denote the merit of document d_i as $M(\text{rel}_i) \geq 0$, or simply M_i, and we state that each document in the candidate set should get exposure proportional to its merit M_i.

$$\forall d_i \in d^q : \text{Exposure}(d_i|\pi) \propto M(\text{rel}_i)$$

For many queries, however, this set of exposure constraints is infeasible. As an example, consider the following example.

Example 1. Consider a query where one document in the candidate set has relevance 1, while all other documents have small relevance ϵ. For sufficiently small ϵ, any ranking will provide too much exposure to the ϵ-relevant documents since we have to put these documents at least somewhere in the ranking. This would however violate the proportional exposure constraint above, and this shortcoming is also present in the Disparate Exposure measure (Section § 3.2.3) and the Equity of Attention constraint of Biega et al. (2018).

Note that this overabundance of exposure for some documents is not a fairness problem, since the extra exposure that some items receive does not come at the expense of other items. Furthermore, it is typically the items that have a slightly lower merit that get disadvantaged by utility maximization, as illustrated in the introductory example. We thus replace the proportionality constraint with the following set of inequality constraints where $\forall d_i, d_j \in d^q$ with $M(\text{rel}_i) \geq M(\text{rel}_j) > 0$,

$$\frac{\text{Exposure}(d_i|\pi)}{M(\text{rel}_i)} \leq \frac{\text{Exposure}(d_j|\pi)}{M(\text{rel}_j)}$$

This one-sided set of constraints still enforce proportionality of exposure to merit but allows the allocation of overabundant exposure, which is achieved by only enforcing that higher-merit items do not get exposure beyond their merit. Note that the opposite direction of the constraint is already encouraged by utility maximization, where high-merit items tend to receive more exposure than they deserve.

Connecting this reasoning back to the example, after putting the item with relevance 1 at rank one, we have to put ϵ-relevant items in position two and further. These ϵ-relevant items are now overexposed which violates the two-sided constraint, but not the one-sided constraint. In this way, the one-sided metric together with utility maximization allows non-relevant items to get higher exposure when this is unavoidable in the tail of the ranking. In the other direction, the metric counteracts unmerited rich-get-richer dynamics, as present in the motivating example earlier.

Measuring disparate exposure. We can now define the following disparity measure $\mathcal{D}$ that captures in how far the fairness-of-exposure constraints are violated

$$\mathcal{D}_{\text{ind}}(\pi|q) = \frac{1}{|H_q|} \sum_{(i,j)\in H_q} \max\left[0, \frac{v_\pi(d_i)}{M_i} - \frac{v_\pi(d_j)}{M_j}\right], \tag{4.3}$$

where $H_q = \{(i, j) \text{ s.t. } M_i \geq M_j > 0\}$. The measure $\mathcal{D}_{\text{ind}}(\pi|q)$ is always non-negative and equals zero only when the individual constraints are exactly satisfied.

Group fairness disparity. The disparity measure from above implements an individual notion of fairness, while other applications ask for a group-based notion. Here, fairness is aggregated over the members of each group. A group of documents can refer to sets of items sold by one seller in an online marketplace, to content published by one publisher, or to job candidates belonging to a protected group. Similar to the case of individual fairness, we want to allocate exposure to groups proportional to their merit. Hence, in the case of only two groups G_0 and G_1, we can define the following group fairness disparity for query q as

$$\mathcal{D}_{\text{group}}(\pi|q) = \max\left(0, \frac{v_\pi(G_i)}{M_{G_i}} - \frac{v_\pi(G_j)}{M_{G_j}}\right), \tag{4.4}$$

where G_i and G_j are such that $M_{G_i} \geq M_{G_j}$ and $\text{Exposure}(G|\pi) = v_\pi(G) = \frac{1}{|G|}\sum_{d_i \in G} v_\pi(d_i)$ is the average exposure of group G, and the merit of the group G is denoted by $M_G = \frac{1}{|G|}\sum_{d_i \in G} M_i$.

4.2 FAIR-PG-RANK: A Policy Learning Algorithm for Fair LTR

In the previous section, we defined a general framework for learning ranking policies under fairness-of-exposure constraints. What remains to be shown is that there exists a stochastic policy class Π and an associated training algorithm that can solve the objective in Equation (4.1) under the disparities $\mathcal{D}$ defined above. To this effect, we now present the FAIR-PG-RANK algorithm. In particular, we first define a class of Plackett-Luce ranking policies that incorporate a machine learning model and then present a policy-gradient approach to efficiently optimize the training objective.

4.2.1 Plackett-Luce Ranking Policies

The ranking policies π we define in the following consist of two components: a scoring model that defines a distribution over rankings, and its associated sampling method. Starting with the scoring model h_θ, we allow any differentiable machine learning model with parameters θ, for example a linear model or a neural network. Given an input $\mathbf{x}^q$ representing the feature vectors of all query-document pairs of the candidate set, the scoring model outputs a vector of scores $h_\theta(\mathbf{x}^q) = (h_\theta(x_1^q), h_\theta(x_2^q), \ldots h_\theta(x_{n_q}^q))$. Based on this score vector, the probability $\pi_\theta(r|q)$ of a ranking $r = \langle r(1), r(2), \ldots r(n_q)\rangle$ under the Plackett-Luce model (Plackett, 1975) is the following product of Softmax distributions

$$\pi_\theta(r|q) = \prod_{i=1}^{n_q} \frac{\exp\left(h_\theta(x_{r(i)}^q)\right)}{\exp\left(h_\theta(x_{r(i)}^q)\right) + \cdots + \exp\left(h_\theta(x_{r(n_q)}^q)\right)}. \tag{4.5}$$

Note that this probability of a ranking can be computed efficiently and that the derivative of $\pi_\theta(r|q)$ and $\log \pi_\theta(r|q)$ exists whenever the scoring model h_θ is differentiable. Sampling a ranking under the Plackett-Luce model is efficient as well. To sample a ranking, starting from the top, documents are drawn recursively from the probability distribution resulting from Softmax over the scores of the remaining documents in the candidate set, until the set is empty.

4.2.2 Policy-Gradient Training Algorithm (PG-Rank)

The next step is to search this policy space Π for a model that maximizes the objective in Equation (4.1). This section proposes a policy-gradient approach (Williams, 1992; Sutton and Barto, 1998), where we use stochastic gradient descent (SGD) updates to iteratively improve our ranking policy. However, since both U and $\mathcal{D}$ are expectations over rankings sampled from π, computing the gradient brute-force is intractable. In this section, we derive the required gradients over expectations as an expectation over gradients. We then estimate this expectation as an average over a finite sample of rankings from the policy to get an approximate gradient.

Conventional LTR methods that maximize user utility are either designed to optimize over a smoothed version of a specific utility metric, such as SVMRank

(Joachims et al., 2009), RankNet (Burges et al., 2005) etc., or use heuristics to optimize over probabilistic formulations of rankings (e.g., SoftRank (Taylor et al., 2008)). Our LTR setup is similar to ListNet (Cao et al., 2007), however, instead of using a heuristic loss function for utility, we present a policy gradient method to directly optimize over both utility and disparity measures. Directly optimizing the ranking policy via policy-gradient learning has two advantages over most conventional LTR algorithms, which optimize upper bounds or heuristic proxy measures. First, our learning algorithm directly optimizes a specified user utility metric and has no restrictions in the choice of the information retrieval (IR) metric. Second, we can use the same policy-gradient approach on our disparity measure $\mathcal{D}$ as well, since it is also an expectation over rankings. Overall, the use of policy-gradient optimization in the space of stochastic ranking policies elegantly handles the non-smoothness inherent in rankings.

4.2.2.1 PG-Rank: Maximizing User Utility

The user utility of a policy π_θ for a query q is defined as $U(\pi|q) = \mathbb{E}_{r\sim\pi_\theta(r|q)}\Delta(r, \mathrm{rel}^q)$. Note that taking the gradient w.r.t. θ over this expectation is not straightforward, since the space of rankings is exponential in cardinality. To overcome this, we use sampling via the log-derivative trick pioneered in the REINFORCE algorithm (Williams, 1992) as follows:

$$\nabla_\theta U(\pi_\theta|q) = \nabla_\theta \mathbb{E}_{r\sim\pi_\theta(r|q)}\Delta(r, \mathrm{rel}^q) = \mathbb{E}_{r\sim\pi_\theta(r|q)}[\nabla_\theta \log \underbrace{\pi_\theta(r|q)}_{\text{Eq. (4.5)}}\Delta(r, \mathrm{rel}^q)] \tag{4.6}$$

This transformation exploits that the gradient of the expected value of the metric Δ over rankings sampled from π can be expressed as the expectation of the gradient of the log probability of each sampled ranking multiplied by the metric value of that ranking. The final expectation is approximated via Monte-Carlo sampling from the Plackett-Luce model in Equation (4.5).

Note that this policy-gradient approach to LTR, which we call PG-RANK, is novel in itself and beyond fairness. It can be used as a standalone LTR algorithm for virtually any choice of utility metric Δ, including NDCG, DCG, ERR, and Average-Rank. Furthermore, PG-RANK also supports non-linear metrics, IPS-weighted metrics for partial information feedback (Joachims et al., 2017), and listwise metrics that do not decompose as a sum over individual documents (Zhai et al., 2003).

Using baseline for variance reduction. Since making stochastic gradient descent updates with this gradient estimate is prone to high variance, we subtract a baseline term from the reward (Williams, 1992) to act as a control variate for variance reduction. Specifically, in the gradient estimate in Equation (4.6), we replace $\Delta(r, \mathrm{rel}^q)$ with $\Delta(r, \mathrm{rel}^q) - b(q)$ where $b(q)$ is the average Δ for the current query. Note that adding this bias term does not make the gradient biased (see Williams (1992) for proof).

Entropy Regularization While optimizing over stochastic policies, entropy regularization is used as a method for encouraging exploration to avoid convergence to suboptimal deterministic policies (Mnih et al., 2016; Williams and Peng, 1991). For our algorithm, we add the entropy of the probability distribution $\mathrm{Softmax}(h_\theta(\mathbf{x}^q))$ times a regularization coefficient γ to the objective.

4.2.2.2 Minimizing disparity

In this section, we will derive the gradients for utility U and disparity ($\mathcal{D}_{\mathrm{group}}$ and $\mathcal{D}_{\mathrm{ind}}$). Since both U and $\mathcal{D}$ are expectations over rankings sampled from π, computing the gradient brute-force is intractable. Hence, similar to utility optimization, we derive the required gradient over expectations as an expectation over gradients. We then estimate this expectation as an average over a finite sample of rankings from the policy to calculate an approximate gradient. Later, we also present a summary of the FAIR-PG-RANK algorithm.

Gradient of Utility measures. To overcome taking a gradient over expectations, we use the log-derivative trick pioneered in the REINFORCE algorithm (Williams, 1992) as follows

$$\begin{aligned}
\nabla_\theta U(\pi_\theta|q) &= \nabla_\theta \mathbb{E}_{r\sim\pi_\theta(r|q)} \Delta(r, \mathrm{rel}^q) \\
&= \nabla_\theta \sum_{r\in\sigma(n_q)} \pi_\theta(r|q)\Delta(r, \mathrm{rel}^q) \\
&= \sum_{r\in\sigma(n_q)} \nabla_\theta[\pi_\theta(r|q)]\Delta(r, \mathrm{rel}^q) \\
&= \sum_{r\in\sigma(n_q)} \pi_\theta(r|q)\nabla_\theta[\log \pi_\theta(r|q)]\Delta(r, \mathrm{rel}^q) \\
&\qquad\qquad\qquad \text{(Log-derivative trick (Williams, 1992))} \\
&= \mathbb{E}_{r\sim\pi_\theta(r|q)}[\nabla_\theta \log \pi_\theta(r|q)\Delta(r, \mathrm{rel}^q)]
\end{aligned}$$

The expectation over $r \sim \pi_\theta(r|q)$ can be computed as an average over a finite sample of rankings from the policy.

Gradient of Disparity functions. The gradient of the disparity measure for individual fairness can be derived as follows:

$$\begin{aligned}
\nabla_\theta \mathcal{D}_{\mathrm{ind}} &= \nabla_\theta \Big[\frac{1}{|H|}\sum_{(i,j)\in H} \max\Big(0, \frac{v_\pi(d_i)}{M_i} - \frac{v_\pi(d_j)}{M_j}\Big)\Big] \quad (H = \{(i,j) \text{ s.t. } M_i \geq M_j\}) \\
&= \nabla_\theta \Big[\frac{1}{|H|}\sum_{(i,j)\in H} \max(0, \mathrm{pdiff}_q(\pi, i, j))\Big] \\
&= \frac{1}{|H|}\sum_{(i,j)\in H} \mathbb{1}[\mathrm{pdiff}_q(\pi, i, j) > 0]\nabla_\theta \mathrm{pdiff}_q(\pi, i, j)
\end{aligned}$$

$$
\begin{aligned}
\nabla_\theta \mathrm{pdiff}_q(\pi, i, j) &= \nabla_\theta\Big[\frac{v_\pi(d_i)}{M_i} - \frac{v_\pi(d_j)}{M_j}\Big] \\
&= \nabla_\theta \mathbb{E}_{r\sim\pi_\theta(r|q)}\Big[\frac{v_r(d_i)}{M_i} - \frac{v_r(d_i)}{M_j}\Big] \\
&= \mathbb{E}_{r\sim\pi_\theta(r|q)}\Big[(\frac{v_r(d_i)}{M_i} - \frac{v_r(d_i)}{M_j})\nabla_\theta \log \pi_\theta(r|q)\Big]
\end{aligned}
$$

(using the log-derivative trick)

The gradient of the disparity measure for group fairness can be derived as follows:

$$\nabla_\theta \mathcal{D}_{\text{group}}(\pi|G_0, G_1, q) = \nabla_\theta \max(0, \xi_q \mathrm{diff}(\pi|q))$$

where $\mathrm{diff}(\pi|q) = \left(\frac{v_\pi(G_0)}{M_{G_0}} - \frac{v_\pi(G_1)}{M_{G_1}}\right)$, and $\xi_q = +1$ if $M_{G_0} \geq M_{G_1}$, -1 otherwise. Further,

$$\nabla_\theta \mathcal{D}_{\text{group}}(\pi|G_0, G_1, q) = \mathbb{1}[\xi_q \mathrm{diff}(\pi|q) > 0]\xi_q \nabla_\theta \mathrm{diff}(\pi|q)$$

(where $\nabla_\theta \mathrm{diff}(\pi_\theta|q) = \nabla_\theta \left[\frac{v_\pi(G_0)}{M_{G_0}} - \frac{v_\pi(G_1)}{M_{G_1}}\right]$)

$$
\begin{aligned}
&= \nabla_\theta\Big[\frac{\frac{1}{|G_0|}\sum_{d\in G_0}\mathbb{E}_{r\sim\pi_\theta} v_r(d)}{\frac{1}{|G_0|}\sum_{d\in G_0} M(\mathrm{rel}_d)} - \frac{\frac{1}{|G_1|}\sum_{d\in G_1}\mathbb{E}_{r\sim\pi_\theta} v_r(d)}{\frac{1}{|G_1|}\sum_{d\in G_1} M(\mathrm{rel}_d)}\Big] \\
&= \nabla_\theta \mathbb{E}_{r\sim\pi_\theta}\left[\frac{\sum_{d\in G_0} v_r(d)}{\sum_{d\in G_0} M(\mathrm{rel}_d)} - \frac{\sum_{d\in G_1} v_r(d)}{\sum_{d\in G_1} M(\mathrm{rel}_d)}\right] \\
&= \mathbb{E}_{r\sim\pi_\theta}\Big[\Big(\frac{\sum_{d\in G_0} v_r(d)}{\sum_{d\in G_0} M(\mathrm{rel}_d)} - \frac{\sum_{d\in G_1} v_r(d)}{\sum_{d\in G_1} M(\mathrm{rel}_d)}\Big)\nabla_\theta \log \pi_\theta(r|q)\Big]
\end{aligned}
$$

Similarly, the expectation over $r \sim \pi_\theta(r|q)$ can be computed as an average over a finite sample of rankings from the policy.

In the derivations above, when a disparity-of-exposure term $\mathcal{D}$ is included in the training objective, we also need to compute the gradient of this term. Fortunately, it has a structure similar to the utility term, so that the same Monte-Carlo approach applies. Specifically, for the **individual-fairness disparity** measure in

Equation (4.3), the gradient can be computed as:

$$\nabla_\theta \mathcal{D}_{\text{ind}} = \frac{1}{|H|} \sum_{(i,j)\in H} \mathbb{1}\Big[\Big(\frac{v_\pi(d_i)}{M_i} - \frac{v_\pi(d_j)}{M_j}\Big) > 0\Big] \times \mathbb{E}_{r\sim\pi_\theta(r|q)}\Big[\big(\frac{v_r(d_i)}{M_i} - \frac{v_r(d_j)}{M_j}\big)\nabla_\theta \log \pi_\theta(r|q)\Big].$$

$$(H = \{(i, j) \text{ s.t. } M_i \geq M_j\})$$

For the **group-fairness disparity** measure defined in Equation (4.4), the gradient can be derived as follows:

$$\nabla_\theta \mathcal{D}_{\text{group}}(\pi|G_0, G_1, q) = \nabla_\theta \max(0, \xi_q \text{diff}(\pi|q)) = \mathbb{1}[\xi_q \text{diff}(\pi|q) > 0]\xi_q \nabla_\theta \text{diff}(\pi|q)$$

where $\text{diff}(\pi|q) = \left(\frac{v_\pi(G_0)}{M_{G_0}} - \frac{v_\pi(G_1)}{M_{G_1}}\right)$, and $\xi_q = \text{sign}(M_{G_0} - M_{G_1})$.

$$\nabla_\theta \text{diff}(\pi|q) = \mathbb{E}_{r\sim\pi_\theta}\Big[\Big(\frac{\sum_{d\in G_0} v_r(d)}{\sum_{d\in G_0} M(\text{rel}_d)} - \frac{\sum_{d\in G_1} v_r(d)}{\sum_{d\in G_1} M(\text{rel}_d)}\Big)\nabla_\theta \log \pi_\theta(r|q)\Big]$$

The expectation of the gradient in both the cases can be estimated as an average over a Monte Carlo sample of rankings from the distribution. The size of the sample is denoted by S in the rest of this chapter.

The completes all necessary ingredients for SGD training of objective (4.1), and now we present all steps of the FAIR-PG-RANK algorithm.

4.2.3 Summary of the FAIR-PG-RANK algorithm

Algorithm 1 summarizes our method for learning fair ranking policies given a training dataset.

4.3 Empirical Evaluation

We conduct experiments on simulated and real-world datasets to empirically evaluate our approach. First, in Section §4.3.2, we validate that the policy-

Algorithm 1: FAIR-PG-RANK

Input: $\mathcal{T} = \{(\mathbf{x}^q, \mathrm{rel}^q)\}_{i=1}^{N}$, disparity measure $\mathcal{D}$, utility/fairness trade-off λ
Parameters: model h_θ, learning rate η, entropy reg γ
Initialize h_θ with parameters θ_0
repeat
  $q = (\mathbf{x}^q, \mathrm{rel}^q) \sim \mathcal{T}$ {Draw a query from training set}
  $h_\theta(\mathbf{x}^q) = (h_\theta(x_1^q), h_\theta(x_2^q), \ldots h_\theta(x_{n_q}^q))$ {Obtain scores for each document}
  for $i = 1$ **to** S **do**
    $r_i \sim \pi_\theta(r|q)$ {Plackett-Luce sampling}
  end for
  $\nabla \leftarrow \hat{\nabla}_\theta U - \lambda \hat{\nabla}_\theta \mathcal{D}$ {Compute gradient as an average over all r_i using §4.2.2.1 and §4.2.2.2}
  $\theta \leftarrow \theta + \eta\nabla$ {Update}
until convergence on the validation set

gradient algorithm is competitive with conventional LTR approaches independent of fairness considerations. We accomplish this by comparing our method PG-RANK relative to conventional LTR baselines on the Yahoo! Learning-to-Rank dataset. Second, in Section §4.3.3, we use simulated data to verify that FAIR-PG-RANK can detect and mitigate unfair features. Third, we show the effectiveness of our algorithm on real-world datasets by presenting experiments on the Yahoo! Learning to Rank dataset for individual fairness and the German Credit Dataset (Dheeru and Karra Taniskidou, 2017) for group fairness (Section §4.3.4).

For all the experiments, we use NDCG as the utility metric, define merit using the identity function $M(\mathrm{rel}) = \mathrm{rel}$, and set the position bias $\mathbf{v}$ to follow the same distribution as the gain factor in DCG i.e., $\mathbf{v}_j \propto \frac{1}{log_2(1+j)}$ where $j = 1, 2, 3, \ldots$ is a position in the ranking.

An open-source implementation of FAIR-PG-RANK is available at `https://github.com/ashudeep/Fair-PGRank`.

4.3.1 Experimental Setup

4.3.1.1 Datasets

Yahoo! Learning to Rank dataset. We used SET 1 from the Yahoo! Learning to Rank challenge (Chapelle and Chang, 2011), which consists of $19,944$ training queries and $6,983$ queries in the test set. Each query has a variable-sized candidate set of documents that needs to be ranked. There are a total of $473,134$ and $165,660$ documents in training and test set, respectively. The query-document pairs are represented by a 700-dimensional feature vector. For supervision, each query-document pair is assigned an integer relevance judgment from 0 (bad) to 4 (perfect).

German Credit Dataset. The original German Credit dataset (Dheeru and Karra Taniskidou, 2017) consists of 1000 individuals, each described by a feature vector x_i consisting of 20 attributes with both numerical and categorical features, as well as a label rel_i classifying it as creditworthy ($\mathrm{rel}_i = 1$) or not ($\mathrm{rel}_i = 0$). We adapt this binary classification task to a learning-to-rank task in the following way: for each query q, we sample a candidate set of 10 individuals each, randomly sampling irrelevant documents (non-creditworthy individuals) and relevant documents (creditworthy individuals) in the ratio 4:1. Each individual is identified as a member of group G_0 or G_1 based on their gender attribute.

4.3.1.2 Baselines

We compare our method to two methods:

Post-processing fairness constraint on estimated relevances. First, we train a linear regression model on all the training set query-document pairs that predicts their relevances. For each query in the test set, we use the estimated relevances of the documents as an input to a modified version of the linear program from Singh and Joachims (2018) with the disparate exposure constraint for group fairness (section § 5.5). We use the following linear program to find the optimal ranking that satisfies fairness constraints on estimated relevances for a given value of λ:

$$
\begin{aligned}
\mathbf{P}^* = \operatorname{argmax}_{\mathbf{P}} \quad & \mathbf{u}^T\mathbf{P}\mathbf{v} - \lambda\xi \qquad \text{(where } \mathbf{u}_i = 2^{\hat{\text{rel}}_i} - 1 \text{ and } \mathbf{v}_j = \tfrac{1}{\log 1+j}\text{)} \\
\text{s.t.} \quad & \forall j \ \sum_j \mathbf{P}_{ij} = 1 \\
& \qquad \text{(sum of probabilities for each document)} \\
& \forall i \ \sum_i \mathbf{P}_{ij} = 1 \quad \text{(sum of probabilities at each position)} \\
& \forall i, j \quad 0 \le \mathbf{P}_{ij} \le 1 \qquad \text{(valid probabilities)} \\
& M(G_k) \ge M(G_{k'}) \Rightarrow \left(\frac{\sum_{d_i \in G_k} \mathbf{P}_i^T \mathbf{v}}{M(G_k)} - \frac{\sum_{d_i \in G_{k'}} \mathbf{P}_i^T \mathbf{v}}{M(G_{k'})} \right) \ge -\xi \\
& \qquad \text{(Disparate exposure fairness constraint)} \\
& \xi \ge 0
\end{aligned}
$$

Note that the relevances used in the linear program (in $\mathbf{u}$) are estimated relevances. This is one of the reasons that even when using this linear program to minimize disparity, we cannot guarantee that disparity on unseen queries can be reduced to zero. In contrast to Singh and Joachims (2018), rather than solving the exact constraint, we use a λ hyperparameter to control how much unfairness we can allow. For our experiments, we evaluate the performance for values of $\lambda \in [0, 0.2]$ (at $\lambda = 0.2$, for all queries the disparity measure on estimated relevances was reduced to zero).

The linear program outputs an $n_q \times n_q$-sized probabilistic matrix $\mathbf{P}$ representing the probability of each document at each position. We compare the NDCG and $\mathcal{D}_{\text{group}}$ for this probabilistic matrix to other methods in Sections § 4.3.3 and § 4.3.4.

Previous Fair LTR approach. Zehlike and Castillo (2020) use a cross-entropy loss on the top-1 probability of each document to maximize utility. The top-1 probabilities of each document are obtained through a Softmax over scores output by a linear scoring function. The disparity measure is implemented as the squared loss of the difference between the top-1 exposure of the groups G_0 and G_1. Training is done using stochastic gradient descent on the sum of cross-entropy and λ times the disparity measure. For all our experiments with this method, we did not use any regularization, searched for the best learning rate in the range $[10^{-3}, 1]$, and evaluated the performance for $\lambda \in \{0, 1, 10, 10^2, \ldots, 10^6\}$.

4.3.1.3 Model and Training Details

Yahoo! Learning to Rank challenge dataset. We train two different models for experiments in Section § 4.3.2: a linear model and a neural network. The neural network has one hidden layer of size 32 and a ReLU activation function. For training, all the weights were randomly initialized between $(-0.001, 0.001)$ for the linear model and $(-1/\sqrt{32}, 1/\sqrt{32})$ for the neural network. We use an Adam optimizer with a learning rate of 0.001 for the linear model and 5×10^{-5} for the neural network. For both cases, we set the entropy regularization constant to $\gamma = 1.0$, use a baseline, and use a sample size of $S = 10$ to estimate the gradient. Both models are trained for 20 epochs over the training dataset, updating the model one query at a time.

German Credit Dataset. To validate whether FAIR-PG-RANK can also optimize for Group fairness, we used the modified German Credit Dataset from the UCI repository (Section § 4.3.1.1). We train a linear scoring model with Adam, using a fixed learning rate of 0.001 with no regularization, and a sample size $S = 25$, for different values of λ in the range $[0, 25]$. We compare our method to baselines mentioned in Section § 4.3.1.2.

4.3.2 Can PG-RANK learn accurate ranking policies?

To validate that PG-RANK is indeed a highly effective LTR method, we conduct experiments on the Yahoo dataset (Chapelle and Chang, 2011). We use the standard experiment setup on the SET 1 dataset and optimize NDCG using PG-RANK, which is equivalent to finding the optimal policy in Equation (4.1) with $\lambda = 0$.

We train PG-RANK for two kinds of scoring models: a linear model and a neural network (one hidden layer with 32 hidden units and ReLU activation). Details of the models and training hyperparameters are given in the supplementary material. The policy learned by our method is stochastic, however, for the purpose of evaluation in this task, we use the highest probability ranking of the candidate set for each query to compute the average NDCG@10 and ERR (Expected Reciprocal Rank) over all the test set queries. We compare our evaluation scores with two baselines from Chapelle and Chang (2011): a linear RankSVM (Joachims, 2006) and a non-linear regression-based ranker that uses Gradient-boosted Decision Trees (GBDT) (Ye et al., 2009).

Table 4.1 shows that PG-RANK achieves competitive performance compared

	NDCG@10	ERR
RankSVM (Joachims, 2006)	0.75924	0.43680
GBDT (Ye et al., 2009)	0.79013	0.46201
PG-RANK (Linear model)	0.76145	0.44988
PG-RANK (Neural Network)	0.77082	0.45440

Table 4.1: Comparing PG-RANK to the baseline LTR methods from (Chapelle and Chang, 2011) on the Yahoo! dataset.

to the conventional LTR methods. When comparing PG-RANK to RankSVM for linear models, our method outperforms RankSVM in terms of both NDCG@10 and ERR. This verifies that the policy-gradient approach is effective at optimizing utility without having to rely on a possibly lose convex upper bound like RankSVM. PG-RANK with the non-linear neural network model further improves on the linear model. Furthermore, additional parameter tuning and variance-control techniques from policy optimization are likely to boost the performance of PG-RANK further but are outside the scope of this work.

4.3.3 Can FAIR-PG-RANK effectively trade off between utility and fairness?

We designed a synthetic dataset to allow inspection into how FAIR-PG-RANK trades-off between user utility and fairness of exposure. The dataset contains 100 queries with 10 candidate documents each. In expectation, 8 of those documents belong to the majority group G_0 and 2 belong to the minority group G_1. For each document we independently and uniformly draw two values x_1 and x_2 from the interval $(0, 3)$, and set the relevance of the document to $x_1 + x_2$ clipped between 0 and 5. For the documents from the majority group G_0, the features

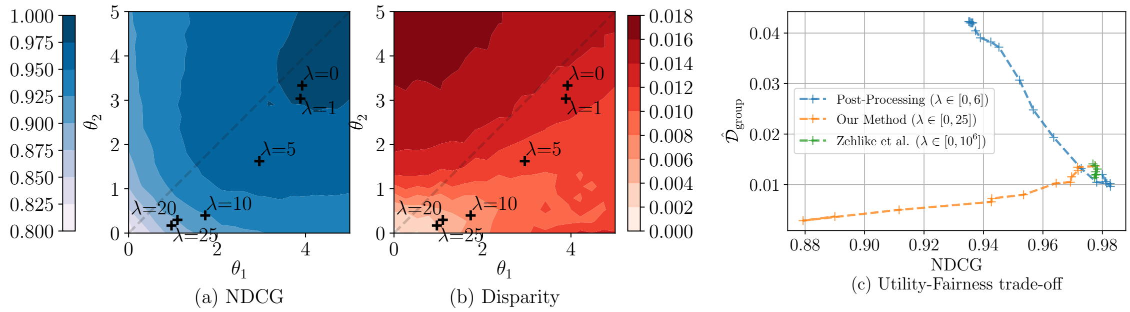


Figure 4.1: Experiments on Simulated dataset. The shaded regions show different ranges of the values of (a) NDCG, (b) Group Disparity ($\mathcal{D}_{\text{group}}$), with varying model parameters $\theta = (\theta_1, \theta_2)$. The (+) points show the models learned by FAIR-PG-RANK under different values of λ. (c) Comparison of NDCG and Group Disparity ($\mathcal{D}_{\text{group}}$) trade-off for different methods.

vector (x_1, x_2) representing the documents provides perfect information about relevance, while for documents in the minority group G_1, however, feature x_2 is corrupted by replacing it with zero so that the information about relevance for documents in G_1 only comes from x_1. This leads to a biased representation between groups, and any use of x_2 is prone to producing unfair exposure between groups.

In order to validate that FAIR-PG-RANK can detect and neutralize this biased feature, we consider a linear scoring model $h_\theta(\mathbf{x}) = \theta_1 x_1 + \theta_2 x_2$ with parameters $\theta = (\theta_1, \theta_2)$. Figure 4.1 shows the contour plots of NDCG and $\mathcal{D}_{\text{group}}$ evaluated for different values of θ. Note that not only the direction of the θ vector affects both NDCG and $\mathcal{D}_{\text{group}}$, but also its length as it determines the amount of stochasticity in π_θ. The true relevance model lies on the $\theta_1 = \theta_2$ line (dotted), however, a fair model is expected to ignore the biased feature x_2. We use PG-RANK to train this linear model to maximize NDCG and minimize $\mathcal{D}_{\text{group}}$. The dots in Figure 4.1 denote the models learned by FAIR-PG-RANK for different values of λ. For small values of λ, FAIR-PG-RANK puts more emphasis on NDCG and thus learns parameter vectors along the $\theta_1 = \theta_2$ direction. As we

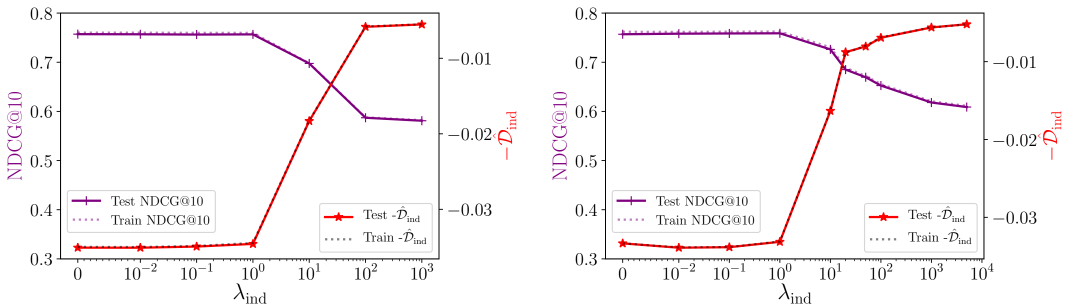


Figure 4.2: Effect of varying λ on NDCG@10 (user utility) and $\mathcal{D}_{\text{ind}}$ (individual fairness disparity) on Yahoo data. *Left*: Linear model, *Right*: Neural Network. The overlapping dotted curves represent the training set NDCG@10 and Disparity, while solid curves show test set performance.

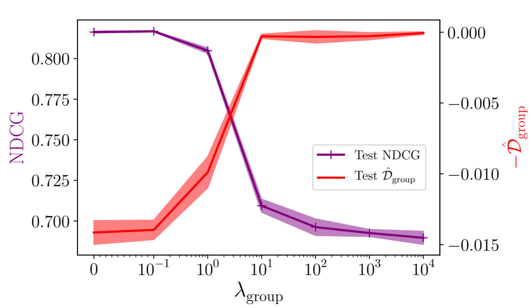
increase emphasis on group fairness disparity $\mathcal{D}_{\text{group}}$ by increasing λ, the policies learned by FAIR-PG-RANK become more stochastic, and it correctly starts to discount the biased attribute by learning models where increasingly $\theta_1 >> \theta_2$.

In Figure 4.1(c), we compare FAIR-PG-RANK with two baselines. As the first baseline, we estimate relevances with a fairness-oblivious linear regression and then use the post-processing method from Singh and Joachims (2018) on the estimates. Unlike FAIR-PG-RANK, which reduces disparity with increasing λ, the post-processing method is misled by the estimated relevances that use the biased feature x_2, and the ranking policies become even less fair as λ is increased. As the second baseline, we apply the method of Zehlike and Castillo (2020), but the heuristic measure it optimizes shows little effect on disparity.

4.3.4 Can FAIR-PG-RANK learn fair ranking policies on real-world data?

In order to study FAIR-PG-RANK on real-world data, we conducted two sets of experiments.

For **Individual Fairness**, we train FAIR-PG-RANK with a linear and a neural network model on the Yahoo! Learning to rank challenge dataset, optimizing Equation 4.1 with different values of λ. The details about the model and training hyperparameters are present in the supplementary material. For both the models, Figure 4.2 shows the average NDCG@10 and $\mathcal{D}_{\text{ind}}$ (individual disparity) over the test and training (dotted line) datasets for different values of λ parameter. As desired, FAIR-PG-RANK emphasizes lower disparity over higher NDCG as the value of λ increases, with disparity going down to zero eventually. Furthermore, the training and test curves for both NDCG and disparity overlap indicating the learning method generalizes to unseen queries. This is expected since both training quantities concentrate around their expectation as the training set size increases.



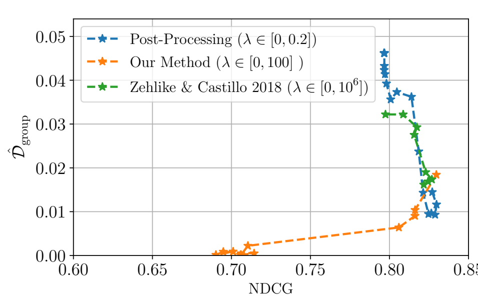


Figure 4.3: *Left*: Effect of varying λ on the test set NDCG and $\mathcal{D}_{\text{group}}$ for the German Credit Dataset. The shaded area shows the standard deviation over five runs of the algorithm on the data. *Right*: Comparison of NDCG and Group Disparity ($\mathcal{D}_{\text{group}}$) trade-off for different methods.

For **Group fairness**, we adapt the German Credit Dataset from the UCI repository (Dheeru and Karra Taniskidou, 2017) to a learning-to-rank task (described in the supplementary), choosing gender as the group attribute. We train FAIR-PG-RANK using a linear model for different values of λ. Figure 4.3 shows that FAIR-PG-RANK is again able to trade off NDCG and fairness effectively. Here we also plot the standard deviation to illustrate that the algorithm reliably converges to solutions of similar performance over multiple runs. Similar to the synthetic example, Figure 4.3 (*right*) again shows that FAIR-PG-RANK can effectively trade-off NDCG for $\mathcal{D}_{\text{group}}$, while the baselines fail.

4.4 Summary

We presented a framework for learning ranking functions that not only maximize utility to their users but that also obey application-specific fairness constraints on how exposure is allocated to the ranked items based on their merit. Based on this framework, we derived the FAIR-PG-RANK policy-gradient algorithm that directly optimizes both utility and fairness without having to resort to upper bounds or heuristic surrogate measures. We demonstrated that our policy-gradient approach is effective for training high-quality ranking functions, that FAIR-PG-RANK can identify and neutralize biased features, and that it can effectively learn ranking functions under both individual fairness and group fairness constraints.

CHAPTER 5
FAIRNESS IN DYNAMIC LEARNING-TO-RANK WITH BIASED FEEDBACK DATA

In the previous chapters, we noted that myopically optimizing utility to the users – as done by virtually all learning-to-rank algorithms – can be unfair to the item providers. We presented a learning-to-rank approach for explicitly enforcing merit-based fairness guarantees to groups of items (e.g., articles by the same publisher, tracks by the same artist). The learning-to-rank algorithm called FAIR-PG-RANK assumes the knowledge of complete relevant judgments at the training time. However, in a real-world scenario, the data is revealed to the training algorithm one query at a time, and even then, it might be incomplete and subject to selection biases because the training data is derived from users who do not equally examine all documents in the ranking.

In this chapter[1], we tackle the problem of learning a ranking function that ensures notions of amortized fairness while simultaneously learning the ranking function from implicit feedback data. The algorithm takes the form of a controller that integrates unbiased estimators for both fairness and utility, dynamically adapting both as more data becomes available. In addition to its rigorous theoretical foundation and convergence guarantees, we find empirically that the algorithm is highly practical and robust.

[1] This chapter is based on joint work with Marco Morik, Jessica Hong, and Thorsten Joachims (Morik et al., 2020). Marco Morik and I contributed equally.

5.1 Introduction

We consider the problem of dynamic Learning-to-Rank (LTR), where the ranking function dynamically adapts based on the feedback that users provide. Such dynamic LTR problems are ubiquitous in online systems — news-feed rankings that adapt to the number of "likes" an article receives, online stores that adapt to the number of positive reviews for a product, or movie-recommendation systems that adapt to who has watched a movie. In all of these systems, learning and prediction are dynamically intertwined, where past feedback influences future rankings in a specific form of online learning with partial information feedback (Cesa-Bianchi and Lugosi, 2006).

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

In this chapter, we present the first dynamic LTR algorithm – called FairCo – that overcomes rich-get-richer dynamics while enforcing a configurable allocation-of-exposure scheme. Unlike existing fair LTR algorithms (Biega et al., 2018; Singh and Joachims, 2018, 2019; Yadav et al., 2021; Zehlike and Castillo, 2020), FairCo explicitly addresses the dynamic nature of the learning problem, where the system is unbiased and fair even though the relevance and the merit of items are still being learned. At the core of our approach lies a merit-based exposure-allocation criterion that is amortized over the learning process (Singh and Joachims, 2018; Biega et al., 2018). We view the enforcement of this merit-based exposure criterion as a control problem and derive a P-controller that optimizes both the fairness of exposure as well as the quality of the rankings. A crucial component of the controller is the ability to estimate merit (e.g., relevance) accurately, even though the feedback is only revealed incrementally as the system operates, and the feedback is biased by the rankings shown in the process (Joachims et al., 2007). To this effect, FairCo includes a new unbiased cardinal relevance estimator – as opposed to existing ordinal methods (Joachims et al., 2017; Agarwal et al., 2019a) –, which can be used both as an unbiased merit estimator for fairness and as a ranking criterion.

In addition to the theoretical justification of FairCo, we provide empirical results on both synthetic news-feed data and real-world movie recommendation data. We find that FairCo is effective at enforcing fairness while providing good ranking performance. Furthermore, FairCo is efficient, robust, and easy to implement.

5.2 Motivation

Consider the following illustrative example of a dynamic LTR problem. An online news-aggregation platform wants to present a ranking of the top news articles on its front page. Through some external mechanism, it identifies a set $\mathcal{D} = \{d_1, ..., d_{20}\}$ of 20 articles at the beginning of each day, but it is left with the learning problem of how to rank these 20 articles on its front page. As users start coming to the platform, the platform uses the following naive algorithm to learn the ranking.

Algorithm 2: Naive Dynamic LTR Algorithm.

Initialize counters $C(d) = 0$ for each $d \in \mathcal{D}$;
foreach *user* **do**
  present ranking $\boldsymbol{\sigma} = \mathrm{argsort}_{\mathcal{D}}[C(d)]$ (random tiebreak);
  increment $C(d)$ for the articles read by the user.

Executing this algorithm at the beginning of a day, the platform starts by presenting the 20 articles in random order for the first user. It may then observe that the user reads the article in position 3 and increments the counter $C(d)$ for this article. For the next user, this article now gets ranked first and the counters are updated based on what the second user reads. This cycle continues for each subsequent user. Unfortunately, this naive algorithm has at least two deficiencies that make it suboptimal or unsuitable for many ranking applications.

The first deficiency lies in the choice of $C(d)$ as an estimate of average relevance for each article – namely the fraction of users that want to read the article. Unfortunately, even with infinite amounts of user feedback, the counters $C(d)$ are not consistent estimators of average relevance (Salganik et al., 2006; Joachims et al., 2007, 2017). In particular, items that happened to get more reads in early iterations get ranked highly, where more users find them and thus have the opportunity to provide more positive feedback for them. This perpetuates a rich-get-richer dynamic, where the feedback count $C(d)$ recorded for each article does not reflect how many users actually wanted to read the article.

The second deficiency of the naive algorithm lies in the ranking policy itself, creating a source of unfairness even if the true average relevance of each article was accurately known (Singh and Joachims, 2018; Biega et al., 2018, 2019). Consider the following omniscient variant of the naive algorithm that ranks the articles by their true average relevance (i.e., the true fraction of users who want to read each article). How can this ranking be unfair? Let us assume that we have two groups of articles, G_{right} and G_{left}, with 10 items each (i.e., articles from politically right- and left-leaning sources). 51% of the users (right-leaning) want to read the articles in group G_{right}, but not the articles in group G_{left}. In reverse, the remaining 49% of the users (left-leaning) like only the articles in G_{left}. Ranking articles solely by their true average relevance puts items from G_{right} into positions 1-10 and the items from G_{left} in positions 11-20. This means the platform gives the articles in G_{left} vastly less exposure than those in G_{right}. We argue that this can be considered unfair since the two groups receive disproportionately different outcomes despite having similar merit (i.e., relevance). Here, a 2% difference in average relevance leads to a much larger difference in exposure between the groups.

We argue that these two deficiencies – namely bias and unfairness – are not just undesirable in themselves, but that they have undesirable consequences. For example, biased estimates lead to poor ranking quality, and unfairness is likely to alienate the left-leaning users in our example, driving them off the platform and encouraging polarization.

Furthermore, note that these two deficiencies are not specific to the news example, but that the naive algorithm leads to analogous problems in many other domains. For example, consider a ranking system for job applicants, where rich-get-richer dynamics and exposure allocation may perpetuate and even amplify existing unfairness (e.g., disparity between male and female applicants). Similarly, consider an online marketplace where products of different sellers (i.e. groups) are ranked. Here rich-get-richer dynamics and unfair exposure allocation can encourage monopolies and drive some sellers out of the market.

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

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

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

With these two desiderata in mind, this chapter develops alternatives to the Naive algorithm. In particular, after introducing the dynamic learning-to-rank setting in Section § 5.4, Section § 5.5 formalizes an amortized notion of merit-based fairness, accounting for the fact that merit itself is unknown at the beginning of the learning process and is only learned throughout. Section § 5.6 then addresses the bias problem, providing estimators that eliminate the presentation bias for both global and personalized ranking policies. Finally, Section § 5.7 proposes a control-based algorithm that is designed to optimize ranking quality while dynamically enforcing fairness.

5.3 Previous Work

Ranking algorithms are widely recognized for their potential for societal impact (Baeza-Yates, 2018), as they form the core of many online systems, including search engines, recommendation systems, news feeds, and online voting. Controlling rich-get-richer phenomena in recommendations and rankings has been studied from the perspective of both optimizing utility through exploration as well as ensuring fairness of such systems (Yin et al., 2012; Schnabel et al., 2016; Abdollahpouri et al., 2017). There are several adverse consequences of naive ranking systems (Ciampaglia et al., 2018), such as political polarization (Beam, 2014), misinformation (Vosoughi et al., 2018), unfair allocation of exposure (Singh and Joachims, 2019), and biased judgment (Baeza-Yates, 2018) through phenomena such as the Matthew effect (Adamic and Huberman, 2000; Germano et al., 2019). Viewing such ranking problems as two-sided markets of users and items that each derive utility from the ranking system brings a novel perspective to tackling such problems (Singh and Joachims, 2018; Abdollahpouri et al., 2019). In this work, we take inspiration from these works to develop methods for mitigating bias and unfairness in a dynamic setting. Apart from the interest in fairness of machine learning algorithms, this chapter also relates to the recent interest in studying the impact of fairness when learning algorithms are applied in dynamic settings (Liu et al., 2018; Ensign et al., 2018; Tabibian et al., 2020).

In information retrieval, there has been a long-standing interest in learning to rank from biased click data. As already argued above, the bias in logged click data occurs because the feedback is incomplete and biased by the presentation. Numerous approaches based on preferences (e.g., (Herbrich et al., 2000;

Joachims, 2002)), click models (e.g., (Chuklin et al., 2015)), and randomized interventions (e.g., (Radlinski and Joachims, 2006)) exist. Most recently, a new approach for debiasing feedback data using techniques from causal inference and missing data analysis was proposed to provably eliminate selection biases (Joachims et al., 2017; Ai et al., 2018). We follow this approach in this work, extend it to the dynamic ranking setting, and propose a new unbiased regression objective in Section § 5.6.

Learning in our dynamic ranking setting is related to the conventional learning-to-rank algorithms such as LambdaRank, LambdaMART, RankNet, Softrank (Burges, 2010; Taylor et al., 2008). However, to implement fairness constraints based on merit, we need to explicitly estimate relevance to the user as a measure of merit while the scores estimated by these methods don't necessarily have a meaning. Our setting is also closely related to online learning to rank for top-k ranking where feedback is observed only on the top-k items, and hence exploration interventions are necessary to ensure convergence (Radlinski et al., 2008; Hofmann et al., 2013; Zoghi et al., 2017; Li et al., 2019). These algorithms are designed with respect to a click-model assumption (Zoghi et al., 2017) or learning in the presence of document features (Li et al., 2019). A key difference in our method is that we do not consider exploration through explicit interventions, but merely exploit user-driven exploration. However, explicit exploration could also be incorporated into our algorithms to improve the convergence rate of our methods.

5.4 Dynamic Learning-to-Rank

We begin by formally defining the dynamic LTR problem. Given is a set of items $\mathcal{D}$ that needs to be ranked in response to incoming requests. At each time step t, a request

$$\mathbf{x}_t, \mathbf{r}_t \sim \mathbf{P}(\mathbf{x}, \mathbf{r}) \tag{5.1}$$

arrives i.i.d. at the ranking system. Each request consists of a feature vector describing the user's information need $\mathbf{x}_t$ (e.g. query, user profile), and the user's vector of true relevance ratings $\mathbf{r}_t$ for all items in the collection $\mathcal{D}$. Only the feature vector $\mathbf{x}_t$ is visible to the system, while the true relevance ratings $\mathbf{r}_t$ are hidden. Based on the information in $\mathbf{x}_t$, a ranking policy $\pi_t(\mathbf{x})$ produces a ranking $\boldsymbol{\sigma}_t$ that is presented to the user. Note that the policy may ignore the information in $\mathbf{x}_t$, if we want to learn a single global ranking like in the introductory news example.

After presenting the ranking $\boldsymbol{\sigma}_t$, the system receives a feedback vector $\mathbf{c}_t$ from the user with a non-negative value $\mathbf{c}_t(d)$ for every $d \in \mathcal{D}$. In the simplest case, it is 1 for click and 0 for no click, and we will use the word "click" as a placeholder throughout this work for simplicity. But the feedback may take many other forms and does not have to be binary. For example, in a video streaming service, the feedback may be the percentage the user watched of each video.

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

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

An instance of such a dynamic LTR algorithm is the Naive algorithm already outlined in Section §5.2. It merely computes $\sum \mathbf{c}_t$ to produce a new ranking policy for $t+1$ (here a global ranking independent of $\mathbf{x}$).

5.4.1 Partial and Biased Feedback

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

$$\mathbf{c}_t(d) = \begin{cases} \mathbf{r}_t(d) & \text{if } \mathbf{e}_t(d) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

Coming back to the running example of news ranking, $\mathbf{r}_t$ contains the full information about which articles the user is interested in reading, while $\mathbf{c}_t$ reveals this information only for the articles d examined by the user (i.e., $\mathbf{e}_t(d) = 1$). Analogously, in the job placement application $\mathbf{r}_t$ indicates for all candidates d whether they are qualified to receive an interview call, but $\mathbf{c}_t$ reveals this information only for those candidates examined by the employer.

A second challenge lies in the fact that the examination vector $\mathbf{e}_t$ cannot be observed. This implies that a feedback value of $\mathbf{c}_t(d) = 0$ is ambiguous – it may either indicate lack of examination (i.e., $\mathbf{e}_t(d) = 0$) or negative feedback (i.e., $\mathbf{r}_t(d) = 0$). This would not be problematic if $\mathbf{e}_t$ was uniformly random, but which items get examined is strongly biased by the ranking $\boldsymbol{\sigma}_t$ presented to the user in the current iteration. Specifically, users are more likely to look at an item high in the ranking than at one that is lower down (Joachims et al., 2007). We model this position bias as a probability distribution on the examination vector

$$\mathbf{e}_t \sim \mathbf{P}(\mathbf{e}\,|\boldsymbol{\sigma}_t, \mathbf{x}_t, \mathbf{r}_t). \tag{5.3}$$

Most click models can be brought into this form (Chuklin et al., 2015). For the simplicity of this work, we merely use the Position-Based Model (PBM) (Craswell et al., 2008). It assumes that the marginal probability of examination $\mathbf{p}_t(d)$ for each item d depends only on the rank $\text{rank}(d|\boldsymbol{\sigma})$ of d in the presented ranking $\boldsymbol{\sigma}$. Despite its simplicity, it was found that the PBM can capture the main effect of position bias accurately enough to be reliable in practice (Joachims et al., 2017; Wang et al., 2018; Agarwal et al., 2019b).

5.4.2 Evaluating Ranking Performance

We measure the quality of a ranking policy π by its utility to the users. Virtually all ranking metrics used in information retrieval define the utility $U(\boldsymbol{\sigma}|\,\mathbf{r})$ of a ranking $\boldsymbol{\sigma}$ as a function of the relevances of the individual items $\mathbf{r}$. In our case, these item-based relevances $\mathbf{r}$ represent which articles the user likes to read, or which candidates are qualified for an interview. A commonly used utility measure is the DCG (Järvelin and Kekäläinen, 2002)

$$U^{DCG}(\boldsymbol{\sigma}|\,\mathbf{r}) = \sum_{d \in \boldsymbol{\sigma}} \frac{\mathbf{r}(d)}{\log_2(1 + \text{rank}(d|\boldsymbol{\sigma}))},$$

or the NDCG when normalized by the DCG of the optimal ranking. Over a distribution of requests $\mathbf{P}(\mathbf{x}, \mathbf{r})$, a ranking policy $\pi(\mathbf{x})$ is evaluated by its expected utility

$$\mathrm{U}(\pi) \;=\; \int U(\pi(\mathbf{x})|\,\mathbf{r})\, d\mathbf{P}(\mathbf{x}, \mathbf{r}). \tag{5.4}$$

5.4.3 Optimizing Ranking Performance

The user-facing goal of dynamic LTR is to converge to the policy $\pi^* = \operatorname{argmax}_\pi \mathrm{U}(\pi)$ that maximizes utility. Even if we solve the problem of estimating $\mathrm{U}(\pi)$ despite our lack of knowledge of $\mathbf{e}$, this maximization problem could be computationally challenging, since the space of ranking policies is exponential even when learning just a single global ranking. Fortunately, it is easy to show (Robertson, 1977) that sorting-based policies

$$\pi(\mathbf{x}) \equiv \operatorname*{argsort}_{d\in\mathcal{D}} \left[R(d|\mathbf{x})\right], \tag{5.5}$$

where

$$R(d|\mathbf{x}) = \int \mathbf{r}(d)\, d\mathbf{P}(\mathbf{r}\,|\mathbf{x}), \tag{5.6}$$

are optimal for virtually all $U(\boldsymbol{\sigma}|\,\mathbf{r})$ commonly used in IR (e.g. DCG). So, the problem lies in estimating the expected relevance $R(d|\mathbf{x})$ of each item d conditioned on $\mathbf{x}$. When learning a single global ranking, this further simplifies to estimating the expected average relevance $R(d) = \int \mathbf{r}(d)\, d\mathbf{P}(\mathbf{r}, \mathbf{x})$ for each item d. The global ranking can then be derived via

$$\boldsymbol{\sigma} = \operatorname*{argsort}_{d\in\mathcal{D}} \left[R(d)\right] \tag{5.7}$$

In Section § 5.6, we will use techniques from causal inference and missing-data analysis to design unbiased and consistent estimators for $R(d|\mathbf{x})$ and $R(d)$ that only require access to the observed feedback $\mathbf{c}_t$.

5.5 Fairness in Dynamic LTR

While sorting by $R(d|\mathbf{x})$ (or $R(d)$ for global rankings) may provide optimal utility to the user, the introductory example has already illustrated that this ranking can be unfair. There is a growing body of literature to address this unfairness in ranking, and we now extend merit-based fairness to the dynamic LTR setting.

The key scarce resource that a ranking policy allocates among the items is exposure. Based on the model introduced in the previous section, we define the exposure of an item d as the marginal probability of examination $\mathbf{p}_t(d) = \mathbf{P}(\mathbf{e}_t(d) = 1|\boldsymbol{\sigma}_t, \mathbf{x}_t, \mathbf{r}_t)$. It is the probability that the user will see d and thus have the opportunity to read that article, buy that product, or interview that candidate. We discuss in Section §5.6 how to estimate $\mathbf{p}_t(d)$. Taking a group-based approach to fairness, we aggregate exposure by groups $\mathcal{G} = \{G_1, \ldots, G_m\}$.

$$Exp_t(G_i) \quad = \quad \frac{1}{|G_i|} \sum_{d \in G_i} \mathbf{p}_t(d). \tag{5.8}$$

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

In order to formulate fairness criteria that relate exposure to merit, we define the merit of an item as its expected average relevance $R(d)$ and again aggregate over groups.

$$Merit(G_i) = \frac{1}{|G_i|} \sum_{d \in G_i} R(d) \tag{5.9}$$

In Section §5.6, we will discuss how to get unbiased estimates of $Merit(G_i)$ using the biased feedback data $\mathbf{c}_t$.

With these definitions in hand, we can address the types of disparities identified in Section §5.2. Specifically, we extend the Disparity of Exposure criterion from Chapter 3 to the dynamic ranking problem, using an amortized notion of fairness as in (Biega et al., 2018). In particular, for any two groups G_i and G_j the disparity

$$D_\tau^E(G_i, G_j) = \frac{\frac{1}{\tau}\sum_{t=1}^{\tau} Exp_t(G_i)}{Merit(G_i)} - \frac{\frac{1}{\tau}\sum_{t=1}^{\tau} Exp_t(G_j)}{Merit(G_j)} \tag{5.10}$$

measures in how far amortized exposure over τ time steps was fulfilled. This **exposure-based fairness disparity** expresses in how far, averaged over all time steps, each group of items got exposure proportional to its relevance. The further the disparity is from zero, the greater is the violation of fairness. Note that other allocation strategies beyond proportionality could be implemented as well by using alternate definitions of disparity (Chapter 3).

Exposure can also be allocated based on other fairness criteria, for example, a Disparity of Impact that a specific exposure allocation implies (Singh and Joachims, 2018). If we consider the feedback $\mathbf{c}_t$ (e.g., clicks, purchases, votes) as a measure of impact

$$Imp_t(G_i) \quad = \quad \frac{1}{|G_i|}\sum_{d \in G_i} \mathbf{c}_t(d), \tag{5.11}$$

then keeping the following disparity close to zero controls how exposure is allocated to make impact proportional to relevance.

$$D_\tau^I(G_i, G_j) = \frac{\frac{1}{\tau}\sum_{t=1}^{\tau} Imp_t(G_i)}{Merit(G_i)} - \frac{\frac{1}{\tau}\sum_{t=1}^{\tau} Imp_t(G_j)}{Merit(G_j)} \tag{5.12}$$

We refer to this as the **impact-based fairness disparity**. In Section § 5.7 we will derive a controller that drives such exposure and impact disparities to zero.

5.6 Unbiased Estimators

To be able to implement the ranking policies in Equation (5.5) and the fairness disparities in Equations (5.10) and (5.12), we need accurate estimates of the position bias $\mathbf{p}_t$, the expected conditional relevances $R(d|\mathbf{x})$, and the expected average relevances $R(d)$. We consider these estimation problems in the following.

5.6.1 Estimating the Position Bias

Learning a model for $\mathbf{p}_t$ is not part of our dynamic LTR problem, as the position-bias model is merely an input to our dynamic LTR algorithms. Fortunately, several techniques for estimating position-bias models already exist in the literature (Joachims et al., 2017; Wang et al., 2018; Agarwal et al., 2019b; Fang et al., 2019), and we are agnostic to which of these is used. In the simplest case, the examination probabilities $\mathbf{p}_t(d)$ only depend on the rank of the item in $\boldsymbol{\sigma}$, analogous to a Position-Based Click Model (Craswell et al., 2008) with a fixed probability for each rank. It was shown in (Joachims et al., 2017; Wang et al., 2018; Agarwal et al., 2019b) how these position-based probabilities can be estimated from explicit and implicit swap interventions. Furthermore, it was shown in (Fang et al., 2019) how the contextual features $\mathbf{x}$ about the users and query can be incorporated in a neural-network based propensity model, allowing it to capture that certain users may explore further down the ranking for some queries. Once any of these propensity models are learned, they can be applied to predict $\mathbf{p}_t$ for any new query $\mathbf{x}_t$ and ranking $\boldsymbol{\sigma}_t$.

5.6.2 Estimating Conditional Relevances

The key challenge in estimating $R(d|\mathbf{x})$ from Equation (5.6) lies in our inability to directly observe the true relevances $\mathbf{r}_t$. Instead, the only data we have is the partial and biased feedback $\mathbf{c}_t$. To overcome this problem, we take an approach inspired by (Joachims et al., 2017) and extend it to the dynamic ranking setting. The key idea is to correct for the selection bias with which relevance labels are observed in $\mathbf{c}_t$ using techniques from survey sampling and causal inference (Horvitz and Thompson, 1952; Imbens and Rubin, 2015). However, unlike the ordinal estimators proposed in (Joachims et al., 2017), we need cardinal relevance estimates since our fairness disparities are cardinal in nature. We, therefore, propose the following cardinal relevance estimator.

The key idea behind this estimator lies in a training objective that only uses $\mathbf{c}_t$, but that in expectation is equivalent to a least-squares objective that has access to $\mathbf{r}_t$. To start the derivation, let's consider how we would estimate $R(d|\mathbf{x})$, if we had access to the relevance labels $(\mathbf{r}_1, ..., \mathbf{r}_\tau)$ of the previous τ time steps. A straightforward solution would be to solve the following least-squares objective for a given regression model $\hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)$ (e.g., a neural network), where w are the parameters of the model.

$$\mathcal{L}^{\mathbf{r}}(w) \;=\; \sum_{t=1}^{\tau}\sum_{d}\left(\mathbf{r}_t(d) - \hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)\right)^2 \tag{5.13}$$

The minimum w^* of this objective is the least-squares regression estimator of $R(d|\mathbf{x}_t)$. Since the $(\mathbf{r}_1, ..., \mathbf{r}_\tau)$ are not available, we define an asymptotically equivalent objective that merely uses the biased feedback $(\mathbf{c}_1, ..., \mathbf{c}_\tau)$. The new objective corrects for the position bias using Inverse Propensity Score (IPS) weighting (Horvitz and Thompson, 1952; Imbens and Rubin, 2015), where the position bias $(\mathbf{p}_1, ..., \mathbf{p}_\tau)$ takes the role of the missingness model.

$$\mathcal{L}^{\mathbf{c}}(w) \;=\; \sum_{t=1}^{\tau}\sum_{d}\hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)^2 + \frac{\mathbf{c}_t(d)}{\mathbf{p}_t(d)}(\mathbf{c}_t(d) - 2\hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)) \tag{5.14}$$

We denote the regression estimator defined by the minimum of this objective as $\hat{R}^{\mathrm{Reg}}(d|\mathbf{x}_t)$. The regression objective in (5.14) is unbiased, meaning that its expectation is equal to the regression objective $\mathcal{L}^{\mathbf{r}}(w)$ that uses the unobserved true relevances $(\mathbf{r}_1, ..., \mathbf{r}_\tau)$.

$$
\begin{aligned}
\mathbb{E}_{\mathbf{e}}\left[\mathcal{L}^{\mathbf{c}}(w)\right] &= \sum_{t=1}^{\tau} \sum_{d} \sum_{\mathbf{e}_t(d)} \left[\hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)^2 + \frac{\mathbf{c}_t(d)}{\mathbf{p}_t(d)} (\mathbf{c}_t(d) - 2\hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)) \right] \mathbf{P}(\mathbf{e}_t(d)|\boldsymbol{\sigma}_t, \mathbf{x}_t) \\
&= \sum_{t=1}^{\tau} \sum_{d} \hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)^2 + \frac{1}{\mathbf{p}_t(d)}\, \mathbf{r}_t(d)(\mathbf{r}_t(d) - 2\hat{R}^{\mathrm{w}}(d|\mathbf{x}_t))\, \mathbf{p}_t(d) \\
&= \sum_{t=1}^{\tau} \sum_{d} \hat{R}^{\mathrm{w}}(d|\mathbf{x}_t)^2 + \mathbf{r}_t(d)^2 - 2\, \mathbf{r}_t(d) \hat{R}^{\mathrm{w}}(d|\mathbf{x}_t) \\
&= \sum_{t=1}^{\tau} \sum_{d} \left(\mathbf{r}_t(d) - \hat{R}^{\mathrm{w}}(d|\mathbf{x}_t) \right)^2 \\
&= \mathcal{L}^{\mathbf{r}}(w)
\end{aligned}
$$

Line 2 formulates the expectation in terms of the marginal exposure probabilities $\mathbf{P}(\mathbf{e}_t(d)|\boldsymbol{\sigma}_t, \mathbf{x}_t)$, which decomposes the expectation as the objective is additive in d. Note that $\mathbf{P}(\mathbf{e}_t(d) = 1|\boldsymbol{\sigma}_t, \mathbf{x}_t)$ is therefore equal to $\mathbf{p}_t(d)$ under our exposure model. Line 3 substitutes $\mathbf{c}_t(d) = \mathbf{e}_t(d)\, \mathbf{r}_t(d)$ and simplifies the expression, since $\mathbf{e}_t(d)\, \mathbf{r}_t(d) = 0$ whenever the user is not exposed to an item. Note that the propensities $\mathbf{p}_t(\boldsymbol{\sigma})$ for the exposed items now cancel, as long as they are bounded away from zero – meaning that all items have some probability of being found by the user. In case users do not naturally explore low enough in the ranking, active interventions can be used to stochastically promote items in order to ensure non-zero examination propensities (e.g. (Hofmann et al., 2013)). Note that unbiasedness holds for any sequence of $(\mathbf{x}_1, \mathbf{r}_1, \boldsymbol{\sigma}_1)..., (\mathbf{x}_T, \mathbf{r}_T, \boldsymbol{\sigma}_T)$, no matter how complex the dependencies between the rankings $\boldsymbol{\sigma}_t$ are.

Beyond this proof of unbiasedness, it is possible to use standard concentration inequalities to show that $\mathcal{L}^{\mathbf{c}}(w)$ converges to $\mathcal{L}^{\mathbf{r}}(w)$ as the size τ of the training sequence increases. Thus, under standard conditions on the capacity for uniform convergence, it is possible to show convergence of the minimizer of $\mathcal{L}^{\mathbf{c}}(w)$ to the least-squares regressor as the size τ of the training sequence increases. We will use this regression objective to learn neural-network rankers in Section § 5.8.2.

5.6.3 Estimating Average Relevances

The conditional relevances $R(d|\mathbf{x})$ are used in the ranking policies from Equation (5.5). But when defining merit in Equation (5.9) for the fairness disparities, the average relevance $R(d)$ is needed. Furthermore, $R(d)$ serves as the ranking criterion for global rankings in Equation (5.7). While we could marginalize $R(d|\mathbf{x})$ over $\mathbf{P}(\mathbf{x})$ to derive $R(d)$, we argue that the following is a more direct way to get an unbiased estimate.

$$\hat{R}^{\text{IPS}}(d) \quad = \quad \frac{1}{\tau}\sum_{t=1}^{\tau}\frac{\mathbf{c}_t(d)}{\mathbf{p}_t(d)}. \tag{5.15}$$

The following shows that this estimator is unbiased as long as the propensities are bounded away from zero.

$$\begin{aligned}
\mathbb{E}_{\mathbf{e}}\left[\hat{R}^{\text{IPS}}(d)\right] &= \frac{1}{\tau}\sum_{t=1}^{\tau}\sum_{\mathbf{e}_t(d)}\frac{\mathbf{e}_t(d)\,\mathbf{r}_t(d)}{\mathbf{p}_t(d)}\mathbf{P}(\mathbf{e}_t(d)|\boldsymbol{\sigma}_t,\mathbf{x}_t) \\
&= \frac{1}{\tau}\sum_{t=1}^{\tau}\frac{\mathbf{r}_t(d)}{\mathbf{p}_t(d)}\,\mathbf{p}_t(d) \\
&= \frac{1}{\tau}\sum_{t=1}^{\tau}\mathbf{r}_t(d) \\
&= R(d)
\end{aligned}$$

In the following experiments, we will use this estimator whenever a direct estimate of $R(d)$ is needed for the fairness disparities or as a global ranking criterion.

5.7 Dynamically Controlling Fairness

Given the formalization of the dynamic LTR problem, our definition of fairness, and our derivation of estimators for all relevant parameters, we are now in the position to tackle the problem of ranking while enforcing the fairness conditions. We view this as a control problem since we need to be robust to the uncertainty in the estimates $\hat{R}(d|\mathbf{x})$ and $\hat{R}(d)$ at the beginning of the learning process. Specifically, we propose a controller that is able to make up for the initial uncertainty as these estimates converge during the learning process.

Following our pairwise definitions of amortized fairness from Section § 5.5, we quantify by how much fairness between all classes is violated using the following overall disparity metric.

$$\overline{D}_\tau = \frac{2}{m(m-1)} \sum_{i=0}^{m} \sum_{j=i+1}^{m} \left| D_\tau(G_i, G_j) \right| \tag{5.16}$$

This metric can be instantiated with the disparity $D^E_\tau(G_i, G_j)$ from Equation (5.10) for exposure-based fairness, or $D^I_\tau(G_i, G_j)$ from Equation (5.12) for impact-based fairness. Since optimal fairness is achieved for $\overline{D}_\tau = 0$, we seek to minimize $\overline{D}_\tau$.

To this end, we now derive a method we call *FairCo*, which takes the form of a Proportional Controller (a.k.a. P-Controller) (Bequette, 2003). A P-controller is a widely used control-loop mechanism that applies feedback through a correction term that is proportional to the error. In our application, the error corresponds to the violation of our amortized fairness disparity from Equations (5.10) and (5.12). Specifically, for any set of disjoint groups $\mathcal{G} = \{G_1, \ldots, G_m\}$, the error term of the controller for any item d is defined as

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

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

Note that the disparity $\hat{D}_{\tau-1}(G_i, G)$ in the error term uses the estimated $\hat{Merit}(G)$ from Equation (5.15), which converges to $Merit(G)$ as the sample size τ increases. To avoid division by zero, $\hat{Merit}(G)$ can be set to some minimum constant.

We are now in a position to state the FairCo ranking policy as

$$\text{FairCo:} \qquad \boldsymbol{\sigma}_\tau = \underset{d \in \mathcal{D}}{\operatorname{argsort}} \left(\hat{R}(d|\mathbf{x}) + \lambda \, \mathbf{err}_\tau(d) \right). \tag{5.17}$$

When the exposure-based disparity $\hat{D}^E_{\tau-1}(G_i, G)$ is used in the error term, we refer to this policy as FairCo(Exp). If the impact-based disparity $\hat{D}^I_{\tau-1}(G_i, G)$ is used, we refer to it as FairCo(Imp).

Like the policies in Section § 5.4.3, FairCo is a sort-based policy. However, the sorting criterion is a combination of relevance $\hat{R}(d|\mathbf{x})$ and an error term representing the fairness violation. The idea behind FairCo is that the error term pushes the items from the underexposed groups upwards in the ranking. The parameter λ can be chosen to be any positive constant. While any choice of λ leads to asymptotic convergence as shown by the theorem below for exposure fairness, a suitable choice of λ can have influence on the finite-sample behavior of FairCo: a higher λ can lead to an oscillating behavior, while a smaller λ makes the convergence smoother but slower. We explore the role of λ in the experiments, but find that keeping it fixed at $\lambda = 0.01$ works well across all of our experiments. Another key quality of FairCo is that it is agnostic to the choice of error metric, and we conjecture that it can easily be adapted to other types of fairness disparities. Furthermore, it is easy to implement and it is very efficient, making it well suited for practical applications.

To illustrate the theoretical properties of FairCo, we now analyze its convergence for the case of exposure-based fairness. To disentangle the convergence of the estimator for $\hat{Merit}(G)$ from the convergence of FairCo, consider a time point τ_0 where $\hat{Merit}(G)$ is already close to $Merit(G)$ for all $G \in \mathcal{G}$. We can thus focus on the question whether FairCo can drive $\overline{D}_\tau^E$ to zero starting from any unfairness that may have persisted at time τ_0. To make this problem well-posed, we need to assume that exposure is not available in overabundance, otherwise it may be unavoidable to give some groups more exposure than they deserve even if they are put at the bottom of the ranking. A sufficient condition for excluding this case is to only consider problems for which the following is true: for all pairs of groups G_i, G_j, if G_i is ranked entirely above G_j at any time point t, then

$$\frac{Exp_t(G_i)}{\hat{Merit}(G_i)} \geq \frac{Exp_t(G_j)}{\hat{Merit}(G_j)}. \tag{5.18}$$

Intuitively, the condition states that ranking G_i ahead of G_j reduces the disparity if G_i has been underexposed in the past. We can now state the following theorem.

Theorem 5.7.1. For any set of disjoint groups $\mathcal{G} = \{G_1, \ldots, G_m\}$ with any fixed target merits $\hat{Merit}(G_i) > 0$ that fulfill (5.18), any relevance model $\hat{R}(d|\mathbf{x}) \in [0, 1]$, any exposure model $\mathbf{p}_t(d)$ with $0 \leq \mathbf{p}_t(d) \leq \mathbf{p}_{\max}$, and any value $\lambda > 0$, running FairCo(Exp) from time τ_0 will always ensure that the overall disparity $\overline{D}_\tau^E$ with respect to the target merits converges to zero at a rate of $O\left(\frac{1}{\tau}\right)$, no matter how unfair the exposures $\frac{1}{\tau_0}\sum_{t=1}^{\tau_0} Exp_t(G_j)$ up to τ_0 have been.

The proof of the theorem is shown in Appendix § 5.A. Note that this theorem holds for any time point τ_0, even if the estimated merits change substantially up to τ_0. So, once the estimated merits have converged to the true merits, FairCo(Exp) will ensure that the amortized disparity $\overline{D}_\tau^E$ converges to zero as well.

5.8 Empirical Evaluation

In addition to the theoretical justification of our approach, we also conducted an empirical evaluation[2]. We first present experiments on a semi-synthetic news dataset to investigate different aspects of the proposed methods under controlled conditions. After that, we evaluate the methods on real-world movie preference data for external validity.

5.8.1 Robustness Analysis on News Data

To be able to evaluate the methods in a variety of specifically designed test settings, we created the following simulation environment from articles in the Ad Fontes Media Bias dataset[3]. It simulates a dynamic ranking problem on a set of news articles belonging to two groups G_{left} and G_{right} (e.g., left-leaning and right-leaning news articles).

In each trial, we sample a set of 30 news articles $\mathcal{D}$. For each article, the dataset contains a polarity value ρ^d that we rescale to the interval between -1 and 1, while the user polarities are simulated. Each user has a polarity that is

---

[2] The implementation is available at `https://github.com/MarcoMorik/Dynamic-Fairness`.

[3] https://www.adfontesmedia.com/interactive-media-bias-chart/

drawn from a mixture of two normal distributions clipped to $[-1, 1]$

$$\rho^{u_t} \sim \text{clip}_{[-1,1]}(p_{neg}\mathcal{N}(-0.5, 0.2) + (1 - p_{neg})\mathcal{N}(0.5, 0.2)) \tag{5.19}$$

where p_{neg} is the probability of the user to be left-leaning (mean=-0.5). We use $p_{neg} = 0.5$ unless specified. In addition, each user has an openness parameter $o^{u_t} \sim \mathcal{U}(0.05, 0.55)$, indicating on the breadth of interest outside their polarity. Based on the polarities of the user u_t and the item d, the true relevance is drawn from the Bernoulli distribution

$$\mathbf{r}_t(d) \sim \text{Bernoulli}\left[p = \exp\left(\frac{-(\rho^{u_t} - \rho^d)^2}{2(o^{u_t})^2}\right)\right].$$

As the model of user behavior, we use the Position-based click model (PBM (Chuklin et al., 2015)), where the marginal probability that user u_t examines an article only depends on its position. We choose an exposure drop-off analogous to the gain function in DCG as

$$\mathbf{p}_t(d) = \frac{1}{\log_2(\text{rank}(d|\boldsymbol{\sigma}_t) + 1)}. \tag{5.20}$$

The remainder of the simulation follows the dynamic ranking setup. At each time step t a user u_t arrives to the system, the algorithm presents an unpersonalized ranking $\boldsymbol{\sigma}_t$, and the user provides feedback $\mathbf{c}_t$ according to $\mathbf{p}_t$ and $\mathbf{r}_t$. The algorithm only observes $\mathbf{c}_t$ and not $\mathbf{r}_t$.

To investigate group-fairness, we group the items according to their polarity, where items with a polarity $\rho^d \in [-1, 0)$ belong to the *left-leaning* group G_{left} and items with a polarity $\rho^d \in [0, 1]$ belong to the *right-leaning* group G_{right}.

We measure ranking quality by the average cumulative NDCG $\frac{1}{\tau}\sum_{t=1}^{\tau} U^{DCG}(\boldsymbol{\sigma}_t|\mathbf{r}_t)$ over all the users up to time τ. We measure Exposure Unfairness via $\overline{D}^E_\tau$ and Impact Unfairness via $\overline{D}^I_\tau$ as defined in Equation (5.16).

In all news experiments, we learn a global ranking and compare the following methods.

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

D-ULTR(Glob): Dynamic LTR by sorting via the unbiased estimates $\hat{R}^{\text{IPS}}(d)$ from Eq. (5.15).

FairCo(Imp): Fairness controller from Eq. (5.17) for impact fairness.

5.8.1.1 Can FairCo reduce unfairness while maintaining good ranking quality?

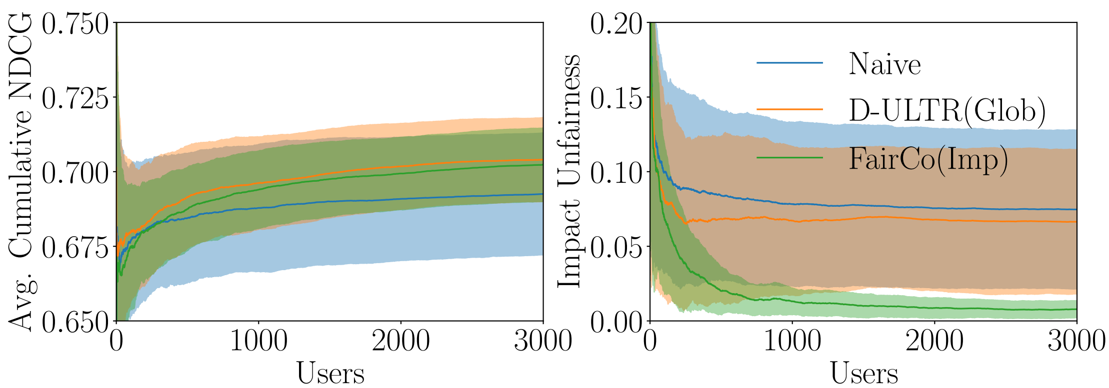


Figure 5.1: Convergence of NDCG (left) and Unfairness (right) as the number of users increases.

This is the key question in evaluating FairCo, and Figure 5.1 shows how NDCG and Unfairness converge for Naive, D-ULTR(Glob), and FairCo(Imp). The plots show that Naive achieves the lowest NDCG and that its unfairness remains high as the number of user interactions increases. D-ULTR(Glob) achieves the best NDCG, as predicted by the theory, but its unfairness is only marginally better than that of Naive. Only FairCo manages to substantially reduce unfairness, and this comes only at a small decrease in NDCG compared to

D-ULTR(Glob).

The following questions will provide further insight into these results, evaluating the components of the FairCo and exploring its robustness.

5.8.1.2 Do the unbiased estimates converge to the true relevances?

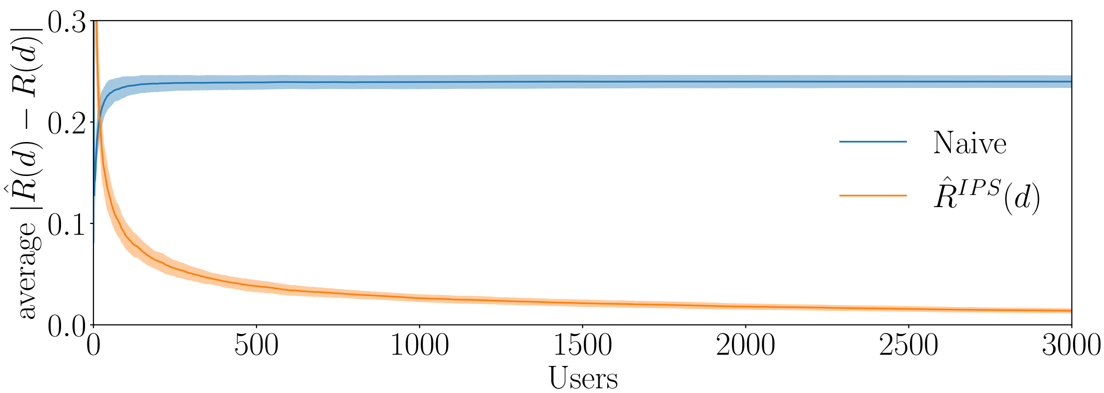


Figure 5.2: Error of relevance estimators as the number of users increases.

The first component of FairCo we evaluate is the unbiased IPS estimator $\hat{R}^{\text{IPS}}(d)$ from Equation (5.15). Figure 1 shows the absolute difference between the estimated global relevance and true global relevance for $\hat{R}^{\text{IPS}}(d)$ and the estimator used in the Naive. While the error for Naive stagnates at around 0.25, the estimation error of $\hat{R}^{\text{IPS}}(d)$ approaches zero as the number of users increases. This verifies that IPS eliminates the effect of position bias and learns accurate estimates of the true expected relevance for each news article so that we can use them for the fairness and ranking criteria.

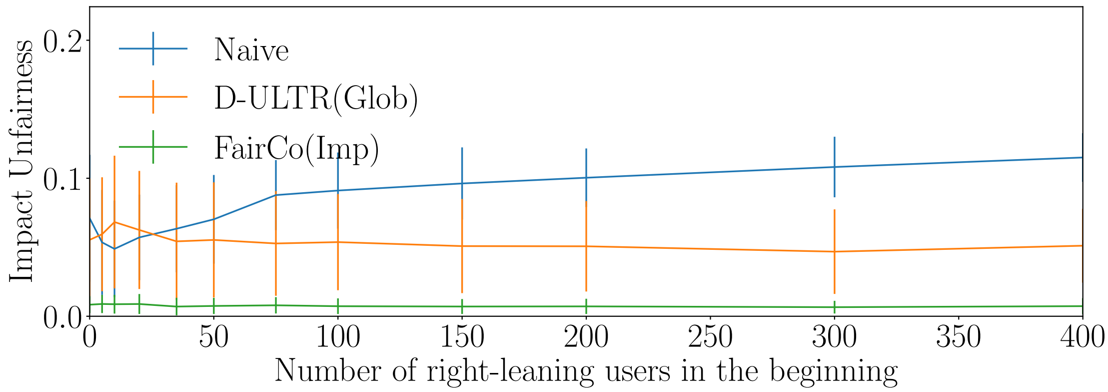


Figure 5.3: The effect of a block of right-leaning users on the Unfairness of Impact.

5.8.1.3 Does FairCo overcome the rich-get-richer dynamic?

The illustrating example in Section §5.2 argues that naively ranking items is highly sensitive to the initial conditions (e.g. which items get the first clicks), leading to a rich-get-richer dynamic. We now test whether FairCo overcomes this problem. In particular, we adversarially modify the user distribution so that the first x users are right-leaning ($p_{neg} = 0$), followed by x left-leaning users ($p_{neg} = 1$), before we continue with a balanced user distribution ($p_{neg} = 0.5$). Figure 5.3 shows the unfairness after 3000 user interactions. As expected, Naive is the most sensitive to the head-start that the right-leaning articles are getting. D-ULTR(Glob) fares better and its unfairness remains constant (but high) independent of the initial user distribution since the unbiased estimator $\hat{R}^{\text{IPS}}(d)$ corrects for the presentation bias so that the estimates still converge to the true relevance. FairCo inherits this robustness to initial conditions since it uses the same $\hat{R}^{\text{IPS}}(d)$ estimator, and its active control for unfairness makes it the only method that achieves low unfairness across the whole range.

5.8.1.4 How effective is the FairCo compared to a more expensive Linear-Programming Baseline?

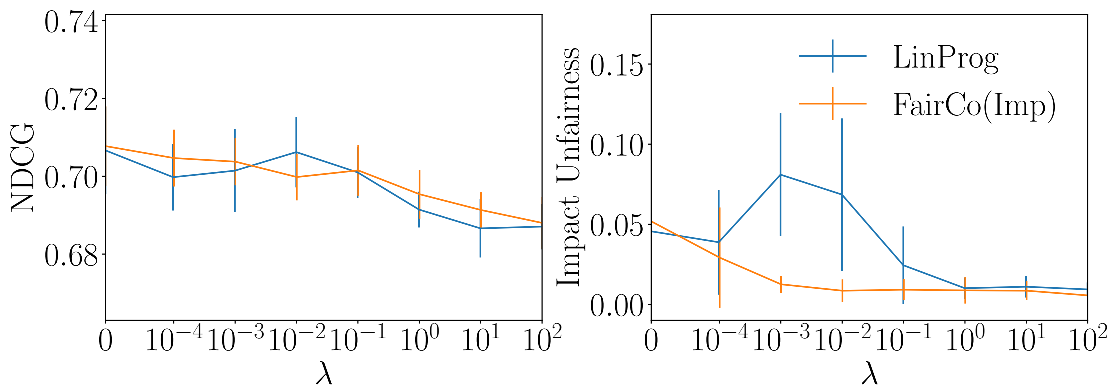


Figure 5.4: Comparing the LP Baseline and the P-Controller in terms of NDCG (left) and Unfairness (right) for different values of λ.

As a baseline, we adapt the linear programming method from (Singh and Joachims, 2018) to the dynamic LTR setting to minimize the amortized fairness disparities that we consider in this work. The method uses the current relevance and disparity estimates to solve a linear programming problem whose solution is a stochastic ranking policy that satisfies the fairness constraints in expectation at each τ. The details of this method are described in Appendix § 5.B. Figure 5.4 shows NDCG and Impact Unfairness after 3000 users averaged over 15 trials for both LinProg and FairCo for different values of their hyperparameter λ. For $\lambda = 0$, both methods reduce to D-ULTR(Glob) and we can see that their solutions are unfair. As λ increases, both methods start enforcing fairness at the expense of NDCG. In these and other experiments, we found no evidence that the LinProg baseline is superior to FairCo. However, LinProg is substantially more expensive to compute, which makes FairCo preferable in practice.

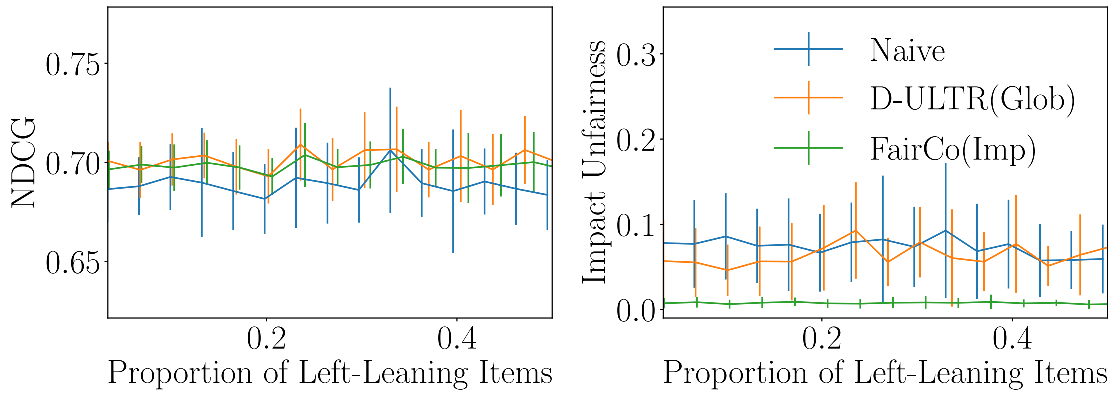


Figure 5.5: NDCG (left) and Unfairness (right) for varying proportion of G_{left}.

5.8.1.5 Is FairCo effective for different group sizes?

In this experiment, we vary asymmetry of the polarity within the set of 30 news articles, ranging from $G_{\text{left}} = 1$ to $G_{\text{left}} = 15$ news articles. For each group size, we run 20 trials for 3000 users each. Figure 5.5 shows that regardless of the group ratio, FairCo reduces unfairness for the whole range while maintaining NDCG. This is in contrast to Naive and D-ULTR(Glob), which suffer from high unfairness.

5.8.1.6 Is FairCo effective for different user distributions?

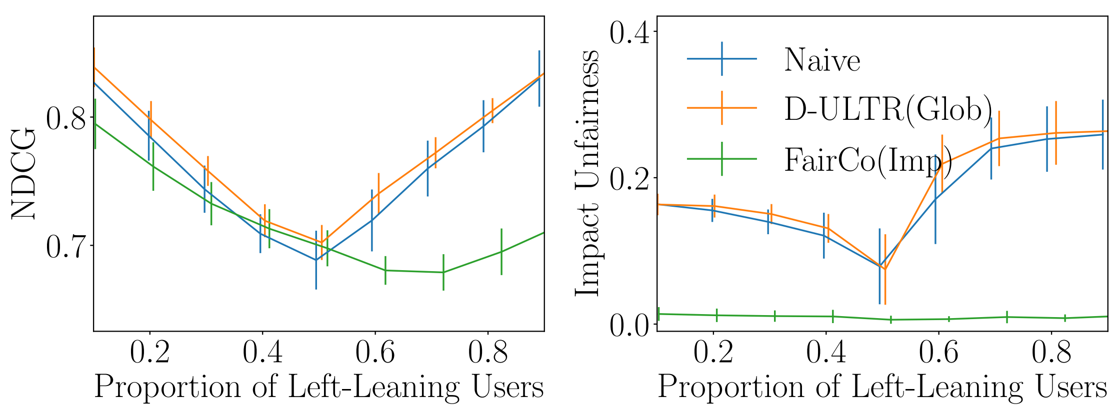


Figure 5.6: NDCG (left) and Unfairness (right) with varying user distributions.

Finally, to examine the robustness to varying user distributions, we control the polarity distribution of the users by varying p_{neg} in Equation (5.19). We run 20 trials each on 3000 users. In Figure 5.6, observe that Naive and D-ULTR(Glob) suffer from high unfairness when there is a large imbalance between the minority and the majority group, while FairCo is able to control the unfairness in all settings.

5.8.2 Evaluation on Real-World Preference Data

To evaluate our method on a real-world preference dataset, we adopt the ML-20M dataset (Harper and Konstan, 2015). We select the five production companies with the most movies in the dataset — *MGM, Warner Bros, Paramount, 20th Century Fox, Columbia*. These production companies form the groups for which we aim to ensure fairness of exposure. To exclude movies with only a few ratings and have a diverse user population, from the set of 300 most rated movies by these production companies, we select 100 movies with the highest standard deviation in the rating across users. For the users, we select 10^4 users who have rated the most number of the chosen movies. This leaves us with a partially filled rating matrix with 10^4 users and 100 movies. To avoid missing data for the ease of evaluation, we use an off-the-shelf matrix factorization algorithm[4] to fill in the missing entries. We then normalize the ratings to $[0, 1]$ by apply a Sigmoid function centered at rating $b = 3$ with slope $a = 10$. These serve as relevance probabilities where higher star ratings correspond to a higher likelihood of positive feedback. Finally, for each trial, we obtain a binary relevance matrix by drawing a Bernoulli sample for each user and movie pair with these

[4] Surprise library (`http://surpriselib.com/`) for SVD with `biased=False` and `D=50`.

probabilities. We use the user embeddings from the matrix factorization model as the user features $\mathbf{x}_t$.

In the following experiments, we use FairCo to learn a sequence of ranking policies $\pi_t(\mathbf{x})$ that are personalized based on $\mathbf{x}$. The goal is to maximize NDCG while providing fairness of exposure to the production companies. User interactions are simulated analogously to the previous experiments. At each time step t, we sample a user $\mathbf{x}_t$ and the ranking algorithm presents a ranking of the 100 movies. The user follows the position-based model from Equation (5.20) and reveal $\mathbf{c}_t$ accordingly.

For the conditional relevance model $\hat{R}^{\text{Reg}}(d|\mathbf{x})$ used by FairCo and D-ULTR, we use one hidden-layer neural network that consists of $D = 50$ input nodes fully connected to 64 nodes in the hidden layer with ReLU activation, which is connected to 100 output nodes with Sigmoid to output the predicted probability of relevance of each movie. Since training this network with less than 100 observations is unreliable, we use the global ranker D-ULTR(Glob) for the first 100 users. We then train the network at $\tau = 100$ users, and then update the network after every 10 users on all previously collected feedback i.e., $\mathbf{c}_1, ..., \mathbf{c}_\tau$ using the unbiased regression objective, $\mathcal{L}^{\mathbf{c}}(w)$, from Eq. (5.14) with the Adam optimizer (Kingma and Ba, 2015).

5.8.2.1 Does personalization via unbiased objective improve NDCG?

We first evaluate whether training a personalized model using the de-biased $\hat{R}^{\text{Reg}}(d|\mathbf{x})$ regression estimator improves ranking performance over a non-personalized model. Figure 5.7 shows that ranking by $\hat{R}^{\text{Reg}}(d|\mathbf{x})$ (i.e. D-ULTR)

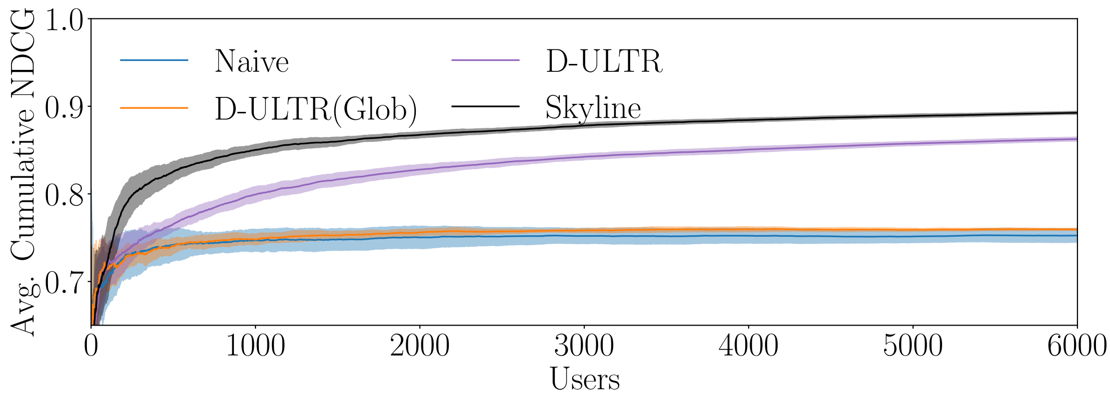


Figure 5.7: Comparing the NDCG of personalized and non-personalized rankings on the Movie data.

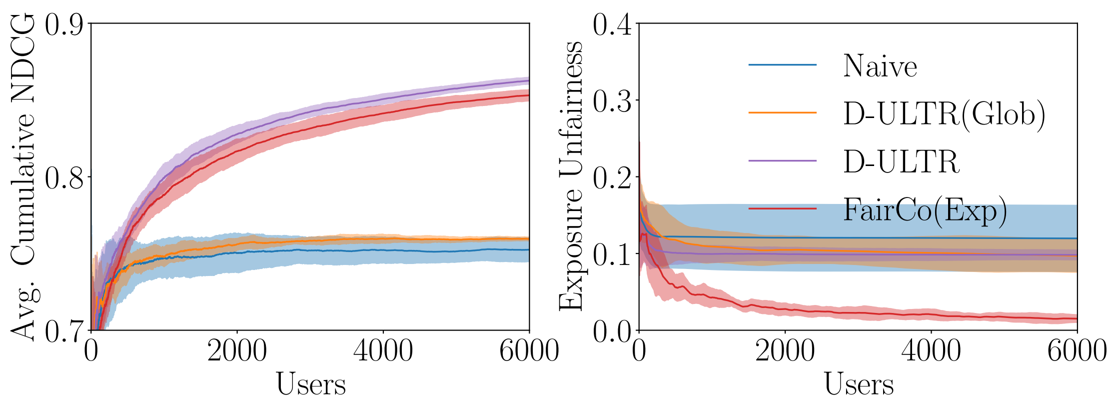


Figure 5.8: NDCG (left) and Exposure Unfairness (right) on the Movie data as the number of user interactions increases.

provides substantially higher NDCG than the unbiased global ranking D-ULTR(Glob) and the Naive ranking. To get an upper bound on the performance of the personalization models, we also train a Skyline model using the (in practice unobserved) true relevances $\mathbf{r}_t$ with the least-squares objective from Eq. (5.13). Even though the unbiased regression estimator $\hat{R}^{\text{Reg}}(d|\mathbf{x})$ only has access to the partial feedback $\mathbf{c}_t$, it tracks the performance of Skyline. As predicted by the theory, they appear to converge asymptotically.

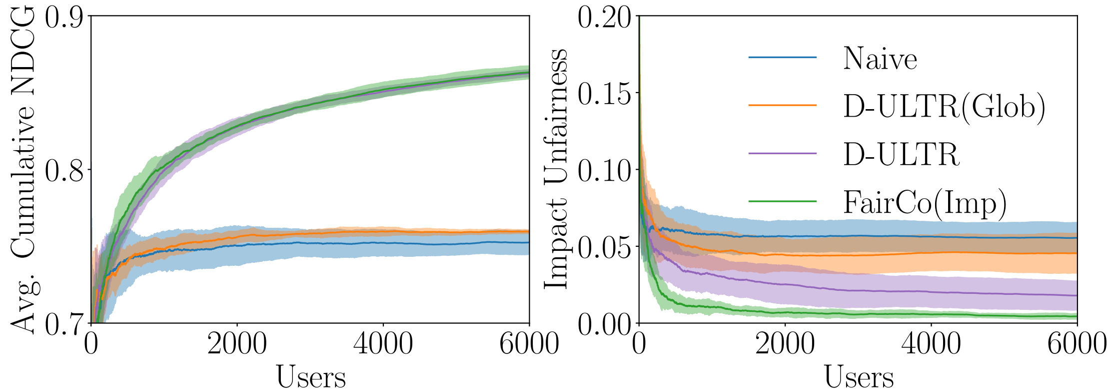


Figure 5.9: NDCG (left) and Impact Unfairness (right) on the Movie data as the number of user interactions increases.

5.8.2.2 Can FairCo reduce unfairness?

Figure 5.8 shows that FairCo(Exp) can effectively control Exposure Unfairness, unlike the other methods that do not actively consider fairness. Similarly, Figure 5.9 shows that FairCo(Imp) is effective at controlling Impact Unfairness. As expected, the improvement in fairness comes at a reduction in NDCG, but this reduction is small.

5.8.2.3 How different are exposure and impact fairness?

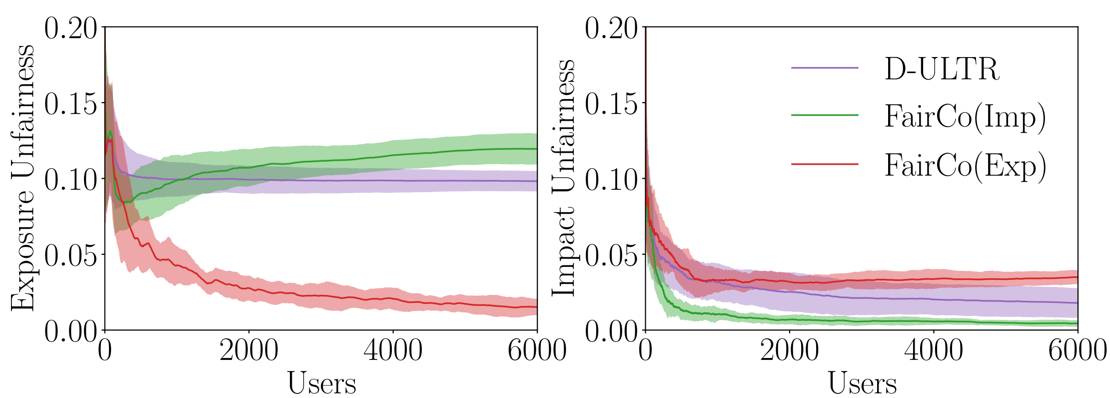


Figure 5.10: Unfairness of Exposure (left) and Unfairness of Impact (right) for the personalized controller optimized for either Exposure or Impact Fairness.

Figure 5.10 evaluates how an algorithm that optimizes Exposure Fairness performs in terms of Impact Fairness and vice versa. The plots show that the two criteria achieve different goals and that they are substantially different. Optimizing for fairness in impact can even increase the unfairness in exposure, illustrating that the choice of criterion needs to be grounded in the application's requirements.

5.9 Summary

In this chapter, we identified how biased feedback and uncontrolled exposure allocation can lead to unfairness and undesirable behavior in dynamic LTR. To address this problem, we proposed FairCo, which can adaptively enforce amortized merit-based fairness constraints even though their underlying relevances are still being learned. The algorithm is robust to presentation bias and thus does not exhibit rich-get-richer dynamics. Moreover, we found that FairCo is easy to implement and computationally efficient, making it well suited for practical applications.

APPENDIX

5.A Convergence of FairCo-Controller

In this section we will prove the convergence theorem of FairCo for exposure fairness. We conjecture that analogous proofs apply to other fairness criteria as well. To prove the main theorem, we will first set up the following lemmas.

Lemma 5.A.1. *Under the conditions of the main theorem, for any value of λ and any $\tau > \tau_0$: if $D^E_{\tau-1}(G_i, G_j) > \frac{1}{(\tau-1)\lambda}$, then*

$$\tau D^E_\tau(G_i, G_j) \leq (\tau - 1) D^E_{\tau-1}(G_i, G_j).$$

Proof. From the definition of D^E_τ in Eq. (5.10) we know that for $\tau > \tau_0$,

$$\tau D^E_\tau(G_i, G_j) = (\tau - 1) D^E_{\tau-1}(G_i, G_j) + \left(\frac{Exp_\tau(G_i)}{\hat{Merit}(G_i)} - \frac{Exp_\tau(G_j)}{\hat{Merit}(G_j)} \right).$$

Since $D^E_{\tau-1}(G_i, G_j) > \frac{1}{(\tau-1)\lambda}$, we know that for all items in G_j it holds that $\mathbf{err}_\tau(d) > \frac{1}{\lambda}$. Hence, FairCo adds a correction term $\lambda \mathbf{err}_\tau(d)$ to the $\hat{R}(d)$ of all $d \in G_j$ that is greater than $\lambda \frac{1}{\lambda} = 1$. Since $0 \leq \hat{R}(d) \leq 1$, the ranking is dominated by the correction term $\lambda \mathbf{err}_\tau(d)$. This means that all $d \in G_j$ are ranked above all $d \in G_i$. Under the feasibility condition from Equation (5.18), this implies that $\left(\frac{Exp_\tau(G_i)}{\hat{Merit}(G_i)} \leq \frac{Exp_\tau(G_j)}{\hat{Merit}(G_j)} \right)$ and thus $\tau D^E_\tau(G_i, G_j) \leq (\tau - 1) D^E_{\tau-1}(G_i, G_j)$. □

Lemma 5.A.2. *Under the conditions of the main theorem, for any value of $\lambda > 0$ there exists $\Delta \geq 0$ such that for any G_i, G_j and $\tau > \tau_0$: if $D^E_{\tau-1}(G_i, G_j) \leq \frac{1}{(\tau-1)\lambda}$, then $\tau D^E_\tau(G_i, G_j) \leq \frac{1}{\lambda} + \Delta$.*

Proof. Using the definition of D^E_τ in Eq. (5.10), we know that

$$\tau D^E_\tau(G_i, G_j) = (\tau - 1) D^E_{\tau-1}(G_i, G_j) + \left(\frac{Exp_\tau(G_i)}{\hat{Merit}(G_i)} - \frac{Exp_\tau(G_j)}{\hat{Merit}(G_j)} \right)$$

$$\leq \frac{1}{\lambda} + \left(\frac{Exp_\tau(G_i)}{\hat{Merit}(G_i)} - \frac{Exp_\tau(G_j)}{\hat{Merit}(G_j)} \right)$$
$$\leq \frac{1}{\lambda} + \Delta$$

where $\Delta = \max_\sigma \max_{\substack{G,G' \\ G \neq G'}} \left(\frac{Exp_\sigma(G)}{\hat{Merit}(G)} - \frac{Exp_\sigma(G')}{\hat{Merit}(G')} \right)$. Note that Δ is a constant independent of τ and refers to the ranking σ for which two groups G, G' have the maximum exposure difference (e.g., one is placed at the top of the ranking, and the other is placed at the bottom). □

Using these two lemmas, we conclude the following theorem:

Theorem 5.A.3. *For any set of disjoint groups $\mathcal{G} = \{G_1, \ldots, G_m\}$ with any fixed target merits $\hat{Merit}(G_i) > 0$ that fulfill* (5.18)*, any relevance model $\hat{R}(d|\boldsymbol{x}) \in [0,1]$, any exposure model $\mathbf{p}_t(d)$ with $0 \leq \mathbf{p}_t(d) \leq \mathbf{p}_{\max}$, and any value $\lambda > 0$, running FairCo(Exp) from time τ_0 will always ensure that the overall disparity $\overline{D}^E_\tau$ with respect to the target merits converges to zero at a rate of $\mathcal{O}\left(\frac{1}{\tau}\right)$, no matter how unfair the exposures $\frac{1}{\tau_0} \sum_{t=1}^{\tau_0} Exp_t(G_j)$ up to τ_0 have been.*

Proof. To prove that $\overline{D}^E_\tau$ converges to zero at a rate of $\mathcal{O}\left(\frac{1}{\tau}\right)$, we will show that for all $\tau \geq \tau_0$, the following holds:

$$\overline{D}^E_\tau \leq \frac{1}{\tau} \frac{2}{m(m-1)} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \max\left(\tau_0 \left| D^E_{\tau_0}(G_i, G_j) \right|, \frac{1}{\lambda} + \Delta \right)$$

The two terms in the max provide an upper bound on the disparity at time τ for any G_i and G_j. To show this, we prove by induction that $\tau D^E_\tau(G_i, G_j) \leq \max\left(\tau_0 \left| D^E_{\tau_0}(G_i, G_j) \right|, \frac{1}{\lambda} + \Delta \right)$ for all $\tau \geq \tau_0$. At the start of the induction at $\tau = \tau_0$, the max directly upper bounds $\tau_0 D^E_{\tau_0}(G_i, G_j)$. In the induction step from $\tau - 1$ to τ, if $(\tau - 1) D^E_{\tau-1}(G_i, G_j) > \frac{1}{\lambda}$, then Lemma 5.A.1 implies that

$$\tau D^E_\tau(G_i, G_j) \leq (\tau - 1) D^E_{\tau-1}(G_i, G_j) \leq \max\left(\tau_0 \left| D^E_{\tau_0}(G_i, G_j) \right|, \frac{1}{\lambda} + \Delta \right).$$

If $(\tau-1)D^E_{\tau-1}(G_i,G_j) \leq \frac{1}{\lambda}$, then Lemma 5.A.2 implies that $\tau D^E_\tau(G_i,G_j) \leq \frac{1}{\lambda} + \Delta \leq \max\left(\tau_0 \left|D^E_{\tau_0}(G_i,G_j)\right|, \frac{1}{\lambda}+\Delta\right)$ as well. This completes the induction, and we conclude that

$$D^E_\tau(G_i,G_j) \leq \frac{1}{\tau}\max\left(\tau_0|D^E_{\tau_0}(G_i,G_j)|, \frac{1}{\lambda}+\Delta\right).$$

Putting everything together, we get

$$\begin{aligned}
\overline{D}^E_\tau &= \frac{2}{m(m-1)}\sum_{i=0}^{m}\sum_{j=i+1}^{m}\left|D^E_\tau(G_i,G_j)\right| \\
&\leq \frac{2}{m(m-1)}\sum_{i=0}^{m}\sum_{j=i+1}^{m}\left|\frac{1}{\tau}\max\left(\tau_0|D^E_{\tau_0}(G_i,G_j)|, \frac{1}{\lambda}+\Delta\right)\right| \\
&\leq \frac{1}{\tau}\frac{2}{m(m-1)}\sum_{i=0}^{m}\sum_{j=i+1}^{m}\max\left(\tau_0\left|D^E_{\tau_0}(G_i,G_j)\right|, \frac{1}{\lambda}+\Delta\right). \qquad \text{(since } \lambda,\Delta,\tau>0\text{)}
\end{aligned}$$

□

5.B Linear Programming Baseline

Here, we present a version of the fairness constraint defined in Singh and Joachims (2018) that explicitly computes an optimal ranking to present in each time step τ by solving a linear program (LP). In particular, we formulate an LP that explicitly maximizes the estimated DCG of the ranking $\boldsymbol{\sigma}_\tau$ while minimizing the estimated cumulative fairness disparity D^I_τ formulated in Equation (5.11). This is used as a baseline to compare with the P-Controller.

To avoid an expensive search over the exponentially-sized space of rankings as in (Biega et al., 2018), we exploit an alternative representation (Singh and Joachims, 2018) as a doubly-stochastic matrix $\mathbb{P}$ that is sufficient for representing $\boldsymbol{\sigma}_t$. In this matrix, the entry $\mathbb{P}_{y,j}$ denotes the probability of placing item y at position j. Both DCG as well as $Imp_\tau(G_i)$ are linear functions of $\mathbb{P}$, which means that the optimum can be computed as the following linear program.

$$\mathbb{P}^* = \underset{\mathbb{P},\xi\geq 0}{\operatorname{argmax}} \underbrace{\sum_y \hat{R}(y|\mathbf{x}) \sum_{j=1}^{n} \frac{\mathbb{P}_{y,j}}{\log(1+j)}}_{\text{Utility}} - \lambda \sum_{i,j} \xi_{ij}$$

$$\text{s.t. } \forall y, j : \sum_{i=1}^{n} \mathbb{P}_{y,i} = 1, \quad \sum_{y'} \mathbb{P}_{y',j} = 1, \quad 0 \leq \mathbb{P}_{y,j} \leq 1$$

$$\forall\, G_i, G_j : \left(\frac{\hat{Imp}_\tau(G_i|\mathbb{P}_\tau)}{\hat{Merit}(G_i)} - \frac{\hat{Imp}_\tau(G_j|\mathbb{P}_\tau)}{\hat{Merit}(G_j)} \right) + D_{\tau-1}(G_i, G_j) \leq \xi_{ij} \tag{5.21}$$

The parameter λ controls trade-off between DCG of $\boldsymbol{\sigma}_t$ and fairness. We explore this parameter empirically in Section § 5.8.1.

It remains to define $\hat{Imp}_\tau(G_j|\mathbb{P}_\tau)$. Assuming the PBM click model with $q(j)$ denoting the examination propensity of item d at position j, the estimated probability of a click is $\hat{R}(d) \cdot q(j)$. So we can estimate the impact on the items in group G for the rankings defined by $\mathbb{P}$ as

$$\hat{Imp}_\tau(G|\mathbb{P}) = \frac{1}{|G|} \sum_{d \in G} \hat{R}(d|\mathbf{x}) \left(\sum_{j=1}^{n} \mathbb{P}_{y,j}\, q(j) \right)$$

We use the `scipy.optimize.linprog` LP solver to solve for the optimal $\mathbb{P}^*$, and then use a Birkhoff von-Neumann decomposition (Birkhoff, 1940) to sample a deterministic ranking $\boldsymbol{\sigma}_\tau$ to present to the user. This ranking is guaranteed to achieve the DCG and fairness optimized by $\mathbb{P}^*$ in expectation.

Note that the number of variables in the LP is $O(n^2 + |\mathcal{G}|^2)$, and even a polynomial-time LP solver incurs substantial computation cost when working with a large number of items in a practical dynamic ranking application.

CHAPTER 6

FAIRNESS IN RANKING UNDER UNCERTAINTY

It is a widely accepted tenet that it is unfair when an agent with higher merit obtains a worse outcome than an agent with lower merit. However, it is hardly ever possible to compute the merit of an agent with certainty. In this chapter[1], our central point is that one of the primary causes of unfairness is uncertainty. A principal or algorithm making decisions never has access to the agents' true merit and instead uses proxy features that only imperfectly predict merit (e.g., GPA, star ratings, recommendation letters). Most of these never fully capture an agent's merit, and yet existing approaches have primarily focused on fairness notions directly based on observed features and outcomes.

In this chapter, we advocate a more principled approach that acknowledges and models the uncertainty explicitly, where the role of observed features is to give rise to a posterior distribution of the agents' merits. We use this viewpoint to define a notion of approximate fairness in ranking. We call an algorithm ϕ-fair (for $\phi \in [0, 1]$) if it has the following property for all agents x and all k: if agent x is among the top k agents with respect to *merit* with probability at least ρ (according to the posterior merit distribution), then the algorithm places the agent among the top k agents in its *ranking* with probability at least $\phi\rho$.

6.1 Introduction

When considering the problem of fair allocation, a fair solution would allocate resources such that if an agent B does not have stronger merits for the resource

[1] This chapter is based on joint work with David Kempe and Thorsten Joachims (Singh et al., 2021).

than A, then B should not get more of the resource than A. Here *merit* could be a qualification (e.g., job performance), a need (e.g., disaster relief), or some other measure of eligibility. However, actual merits are practically always unobservable. Consider the following standard algorithmic decision-making environments: (1) An e-commerce or recommender platform (the principal) displays items (the agents) in response to a user query. An item's merit is the utility the user would derive from it, whereas the platform can only observe numerical ratings, text reviews, the user's past history, and similar features. (2) A job-recommendation site or employer (the principal) wants to recommend/hire one or more applicants (the agents). The merit of an applicant is the applicant's (future) performance on the job over a period of time, whereas the site or employer can only observe (past) grades, test scores, recommendation letters, performance in interviews, and similar assessments.

In both of these examples — and essentially all others in which algorithms are called upon to make allocation decisions between agents — uncertainty about merit is unavoidable, and arises from multiple sources: (1) the training data of a machine learning algorithms is a random sample, (2) the features themselves often come from a random process, and (3) the merit itself may only be revealed in the future after a random process (e.g., whether an item is sold or an employee performs well). Given that decisions will be made in the presence of uncertainty, it is important to define the notion of *fairness* under uncertainty. Extending the aforementioned tenet that "if agent B has less merit than A, then B should not be treated better than A," we state the following generalization to uncertainty about merit, first for just two agents:

Axiom 1. *If A has merit greater than or equal to B with probability at least ρ, then a fair policy should treat A at least as well as B with probability at least ρ.*

This being an axiom, we cannot offer a mathematical justification, but it captures an inherent sense of fairness in the absence of enough information, and it converges to the conventional tenet as uncertainty is reduced. In particular, consider the following situation: two items A, B, with 10 reviews each, have average star ratings of 3.9 and 3.8, respectively; or two job applicants A, B have GPAs of 3.9 and 3.8. While this constitutes some (weak) evidence that A may have more merit than B, this evidence is very uncertain. The posterior merit distributions based on the available information should reflect this uncertainty by having non-trivial variance; our axiom then implies that A and B must be treated similarly to achieve fairness. In particular, it would be highly unfair to *deterministically* rank A ahead of B (or vice versa). Our main point is that this uncertainty, rather than the specific numerical values of 3.9 and 3.8, is what should make a mechanism treat A and B similarly.

Contributions In this chapter, we will study fairness in the presence of uncertainty, specifically for the generalization where the principal must rank n items. The main contribution here is the fairness framework, giving definitions of fairness in ranking in the presence of uncertainty. This framework, including extensions to approximate notions of fairness, is presented in Section § 6.2. We believe that uncertainty of merit is one of the most important sources of unfairness, and modeling it explicitly and axiomatically is key to addressing it.

Next, in Section § 6.3, we present algorithms for a principal to achieve (approximately) fair ranking distributions. A simple way is for the principal to mix between an optimal (unfair) ranking and (perfectly fair) Thompson sampling. However, we show that this policy is not optimal for the principal's utility, and we present an efficient LP-based algorithm that achieves an optimal ranking distribution for the principal, subject to an approximate fairness constraint.

We next explore empirically to what extent a focus on fairness towards the agents reduces the principal's utility. We do so with two extensive sets of experiments: one described in Section §6.5 on existing data, and one described in Section §6.6 "in the wild." In the first set of experiments, we consider movie recommendations based on the MovieLens dataset and investigate to what extent fairness towards movies would result in lower utility for users of the system. The second experiment was carried out at a scientific conference, where we implemented and fielded a paper recommendation system, and half of the conference attendees using the system received rankings that were modified to ensure greater fairness towards papers. We report on various metrics that capture the level of engagement of conference participants based on which group they were assigned to.

The upshot of our experiments and theoretical analysis is that at least in the settings we have studied, high levels of fairness can be achieved at a relatively small loss in utility for the principal and the system's users.

6.2 Ranking with Uncertain Merits

We are interested in ranking policies for a principal (the ranking system, such as an e-commerce platform or a job portal in our earlier examples) whose goal is to rank a set $\mathcal{X}$ of n agents (such as products or applicants). The principal observes some evidence for the merit of the agents and must produce a distribution over rankings trading off fairness to the agents against the principal's utility. For the agents, a higher rank is always more desirable than a lower rank.

6.2.1 Rankings and Ranking Distributions

We use $\Sigma(\mathcal{X})$ to denote the set of all $n!$ rankings, and $\Pi(\mathcal{X})$ for the set of all distributions over $\Sigma(\mathcal{X})$. We express a ranking $\sigma \in \Sigma(\mathcal{X})$ in terms of the agents assigned to given positions, i.e., $\sigma(k)$ is the agent in position k. A ranking distribution $\pi \in \Pi(\mathcal{X})$ can be represented by the $n!$ probabilities $\pi(\sigma)$ of the rankings $\sigma \in \Sigma(\mathcal{X})$. However, all the information relevant for our purposes can be represented more compactly using the *Marginal Rank Distribution*: we write $p_{x,k}^{(\pi)} = \sum_{\sigma:\sigma(k)=x} \pi(\sigma)$ for the probability under π that agent $x \in \mathcal{X}$ is in position k in the ranking. We let $\mathcal{P}^{(\pi)} = (p_{x,k}^{(\pi)})_{x,k}$ denote the $n \times n$ matrix of all marginal rank probabilities.

$\mathcal{P}^{(\pi)}$ is doubly stochastic, i.e., the sum of each row and column is 1. While π uniquely defines $\mathcal{P}^{(\pi)}$, the converse mapping may not be unique. Given a doubly stochastic matrix $\mathcal{P}$, the Birkhoff-von Neumann decomposition (Birkhoff, 1940) can be used to compute *some* ranking distribution π consistent with $\mathcal{P}$, i.e., $\mathcal{P}^{(\pi)} = \mathcal{P}$; any consistent distribution π will suffice for our purposes.

6.2.2 Merit, Uncertainty, and Fairness

The principal must determine a distribution over the rankings of the agents. This distribution will be based on some evidence for the agents' merits. This evidence could take the form of star ratings and reviews of products (combined with the site visitor's partially known preferences), or GPA, test scores, and recommendation letters of an applicant. Our main departure from past work on individual fairness (following (Dwork et al., 2012)) is that we do not view this evidence as having inherent meaning; rather, its sole role is to induce a posterior joint distribution over the agents' merits.

The merit of agent x is $v_x \in \mathbb{R}$, and we write $\boldsymbol{v} = (v_x)_{x \in \mathcal{X}}$ for the vector of all agents' merits. Based on all observed evidence, the principal can infer a distribution Γ over agents' merits using any suitable Bayesian inference procedure. For simplicity of exposition, we assume that the support of $\boldsymbol{v}$ contains no ties, i.e., no two agents will ever have exactly equal merit, for any draw of $\boldsymbol{v}$. We write $\Gamma(\boldsymbol{v})$ for the probability of merits $\boldsymbol{v}$ under Γ. We emphasize that the distribution will typically not be independent over entries of $\boldsymbol{v}$ — for example, students' merit conditioned on observed grades will be correlated via common grade inflation if they took the same class. To avoid awkward tie-breaking issues, we assume that $v_x \neq v_y$ for all distinct $x, y \in \mathcal{X}$ and all $\boldsymbol{v}$ in the support of Γ. These assumptions side-step the need to define the notion of top-k lists with ties, and it comes at little cost in expressivity, as any tie-breaking would typically be encoded in slight perturbations to the v_x anyway.

We write $\mathcal{M}_{x,k}^{(\boldsymbol{v})}$ for the event that under $\boldsymbol{v}$, agent x is among the top k agents with respect to merit, i.e., that $|\{x' \mid v_{x'} > v_x\}| < k$. We now come to our key definition of approximate fairness.

Definition 1 (Approximately Fair Ranking Distribution)**.** *A ranking distribution π is ϕ-fair iff*

$$\sum_{k'=1}^{k} p_{x,k'}^{(\pi)} \geq \phi \cdot \mathbf{P}[\boldsymbol{v} \sim \Gamma] \mathcal{M}_{x,k}^{(\boldsymbol{v})} \tag{6.1}$$

for all agents x and positions k. That is, the ranking distribution π ranks x at position k or above with at least a ϕ fraction of the probability that x is actually among the top k agents according to Γ. *Furthermore, π is* fair *iff it is 1-fair.*

The reason for defining ϕ-approximately fair ranking distributions (rather than just fair distributions) is that fairness typically comes at a cost to the principal (such as lower expected clickthrough or lower expected performance of recommended employees). For example, if the v_x are probabilities that a user will purchase products on an e-commerce site, then deterministically ranking by decreasing $\mathbb{E}_\Gamma [v_x]$ is the principal's optimal ranking under common assumptions about user behavior; yet, being deterministic, it is highly unfair. Our definition of approximate fairness allows, e.g., a policymaker to choose a tradeoff regarding how much fairness (with resulting expected utility loss) to require from the principal. Notice that for $\phi = 0$, the principal is unconstrained.

We remark that in the definition of (approximate) fairness, the numerical values of v_x only matter insofar as comparison is concerned. A useful implication is the following: let $f : \mathbb{R} \to \mathbb{R}$ be any strictly monotone function. If we replace the distribution Γ with a distribution Γ' that draws the vector $(f(v_x))_{x \in \mathcal{X}}$ with probability $\Gamma(\boldsymbol{v})$ (i.e., it replaces each entry v_x with $f(v_x)$), then if π was ϕ-fair with respect to Γ, it is also ϕ-fair with respect to Γ'.

6.2.3 The Principal's Utility

The principal's utility can be the profit of an e-commerce site or the satisfaction of its customers. We assume that the principal's utility for a ranking σ with agent merits $\boldsymbol{v}$ takes the form

$$\mathrm{U}(\sigma \,|\, \boldsymbol{v}) = \sum_{k=1}^{n} w_k v_{\sigma(k)}, \tag{6.2}$$

where w_k is the position weight for position k in the ranking, and we assume that the w_k are non-increasing, i.e., the principal derives the most utility from earlier positions of the ranking.

The assumption that the utility from each position is factorable (i.e., of the form $w_k \cdot v_{\sigma(k)}$) is quite standard in the literature (Järvelin and Kekäläinen, 2002;

Taylor et al., 2008). The assumption that the utility is *linear* in $v_{\sigma(k)}$ is in fact not restrictive at all. To see this, assume that the principal's utility were of the form $w_k \cdot f(v_{\sigma(k)})$ for some strictly increasing function f. As remarked above, the exact same fairness guarantees are achieved when the agents' merits v_x are replaced with $f(v_x)$; doing so preserves fairness, while in fact making the principal's utility linear in the merits. Some common examples of utility functions falling into this general framework are DCG with $w_k = 1/\log_2(1+k)$, Average Reciprocal Rank with $w_k = 1/k$, and Precision@K with $w_k = \mathbb{1}[k \leq K]/K$.

When the ranking and merits are drawn from distributions, the principal's utility is the expected utility under both sources of randomness:

$$\mathrm{U}(\pi \mid \Gamma) = \mathbb{E}_{\sigma\sim\pi, \boldsymbol{v}\sim\Gamma}\left[\mathrm{U}(\sigma \mid \boldsymbol{v})\right]. \tag{6.3}$$

6.2.4 Discussion

Our definition is superficially similar to existing definitions of individual fairness (e.g., (Dwork et al., 2012; Joseph et al., 2016)), in that similar observable features often lead to similar outcomes. Importantly, though, it side-steps the need to define a similarity metric between agents in the feature space. Furthermore, it does not treat the observable attributes (such as star ratings) themselves as any notion of "merit." Instead, our central point is that agents' features should be viewed (solely) as noisy signals about the agents' merits and that a comparison of their merits — and the principal's uncertainty about the merits — should determine the agents' relative ranking. That moves the key task of quantifying individual fairness from articulating which features should be considered relevant for similarity, to articulating what inferences can be drawn about merit from observed features.

One may argue, rightfully, that from an operational perspective, this simply pushes the normative decisions into determining Γ. However, our main point is that this is where normative decisions indeed belong. Rather than trying to define ad hoc notions of fairness on observable features, the principal should determine what the features reveal (and don't reveal) about the agents' merits. In this context, it is worth remarking that the distribution Γ may also take care of systemic biases. For example, if an applicant's features like GPA are biased by particular characteristics (e.g., race or gender), Γ can/should reflect that applicants can have the same merit even if they do not have the same GPA. Again, the important distinction here is to not treat observable features such as GPA or star ratings as merit, but rather as noisy and imperfect (and possibly biased) *signals* about merit.

An additional benefit of requiring the use of fair ranking policies, discussed in depth in Section §6.2.4, is that it makes the principal bear more of the cost of an inaccurate Γ, and thereby incentivizes the principal to improve the distribution Γ. To see this at a high level, notice that if Γ precisely revealed merits, then the optimal and fair policies would coincide. In the presence of uncertainty, an unrestricted principal will optimize utility, and in particular do better than a principal who is constrained to be (partially or completely) fair. Thus, a fair principal stands to gain more by obtaining perfect information.

We emphasize here that we believe that low-stakes scenarios repeated often are typically a better fit for any randomized approach (including ours) than high-stakes one-shot games. The reasons include that for a high-stakes situation, a principal may be less willing to trade off utility for fairness and that for a single decision, verifying that a decision was made probabilistically could be hard or impossible. Thus, ideal scenarios to keep in mind are e-commerce websites or other recommendation systems that repeatedly choose a ranking of products (or papers or other items) in response to a customer's query.

Our work is related to research on defining fairness notions for rankings which can be categorized into three groups: composition-based notions (Yang and Stoyanovich, 2017; Celis et al., 2018; Zehlike et al., 2017), opportunity-based notions (Singh and Joachims, 2018, 2019; Biega et al., 2018), and evidence-based notions (Dwork et al., 2019; Beutel et al., 2019; Narasimhan et al., 2020). Our fairness axiom combines the opportunity-based and evidence-based notions by stating that the economic opportunity allocated to the agents must be consistent with the existing evidence about their relative ordering.

6.3 Optimal and Fair Policies

For a distribution Γ over merits, let σ^*_Γ be the ranking which sorts the agents by expected merit, i.e., by non-increasing $\mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x]$. The following lemma follows because the position weights w_k are non-increasing.

Lemma 6.3.1. *σ^*_Γ is a utility-maximizing ranking policy for the principal.*

Proof. Let π be a randomized policy for the principal. We will use a standard exchange argument to show that making π more similar to σ^* can only increase the principal's utility. Recall that by Equation (6.3), the principal's utility under π can be written as

$$\mathrm{U}(\pi \mid \Gamma) = \sum_{x\in\mathcal{X}} \sum_k p^{(\pi)}_{x,k} \cdot \mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x] \cdot w_k.$$

Assume that π does not sort x by non-increasing $\mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x]$. Then, there exist two positions $j < k$ and two agents x, y such that $\mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x] > \mathbb{E}_{\boldsymbol{v}\sim\Gamma}\left[v_y\right]$, and $p^{(\pi)}_{x,k} > 0$ and $p^{(\pi)}_{y,j} > 0$. Let $\epsilon = \min(p^{(\pi)}_{x,k}, p^{(\pi)}_{y,j}) > 0$, and consider the modified policy which subtracts ϵ from $p^{(\pi)}_{x,k}$ and $p^{(\pi)}_{y,j}$ and adds ϵ to $p^{(\pi)}_{x,j}$ and $p^{(\pi)}_{y,k}$. This changes the expected utility of the policy by

$$\epsilon \cdot (\mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x] \cdot w_j + \mathbb{E}_{\boldsymbol{v}\sim\Gamma}\left[v_y\right] \cdot w_k - \mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x] \cdot w_k - \mathbb{E}_{\boldsymbol{v}\sim\Gamma}\left[v_y\right] \cdot w_j)$$
$$= \epsilon \cdot (w_j - w_k) \cdot (\mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x] - \mathbb{E}_{\boldsymbol{v}\sim\Gamma}\left[v_y\right]) \geq 0.$$

By repeating this type of update, the policy eventually becomes fully sorted, weakly increasing the utility with every step. Thus, the optimal policy must be sorted by $\mathbb{E}_{\boldsymbol{v}\sim\Gamma}[v_x]$. □

While this policy conforms to the Probability Ranking Principle (Robertson, 1977), it violates Axiom 1 for ranking fairness when merits are uncertain. We define a natural solution for a 1-fair ranking distribution based on Thompson Sampling:

Definition 2 (Thompson Sampling Ranking Distribution)**.** *Define $\pi^{\mathrm{TS}}_{\Gamma}$ as follows: first, draw a vector of merits $\boldsymbol{v} \sim \Gamma$, then rank the agents by decreasing merits in $\boldsymbol{v}$.*

That $\pi^{\mathrm{TS}}_{\Gamma}$ is 1-fair follows directly from the definition of fairness. By definition, it ranks each agent x in position k with exactly the probability that x has k-th highest merit.

Lemma 6.3.2. *$\pi^{\mathrm{TS}}_{\Gamma}$ is a 1-fair ranking distribution.*

Proof. Let π be a 1-fair ranking policy. By Equation (6.1), π must satisfy the following constraints:

$$\sum_{k'=1}^{k} p^{(\pi)}_{x,k'} \geq \mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}^{(\boldsymbol{v})}_{x,k} \qquad \text{for all } x \text{ and } k.$$

Summing over all x (for any fixed k), both the left-hand side and right-hand side sum to k; for the left-hand side, this is the expected number of agents placed in the top k positions by π, while for the right-hand side, it is the expected number of agents among the top k in merit. Because the weak inequality above holds for all x and k, yet the sum over x is equal, *each* inequality must hold with *equality*:

$$\sum_{k'=1}^{k} p_{x,k'}^{(\pi)} = \mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}_{x,k}^{(\boldsymbol{v})} \qquad \text{for all } x \text{ and } k.$$

This implies that

$$p_{x,k}^{(\pi)} = \mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}_{x,k}^{(\boldsymbol{v})} - \mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}_{x,k-1}^{(\boldsymbol{v})},$$

which is completely determined by Γ. Substituting these values of $p_{x,k}^{(\pi)}$ into the principal's utility, we see that it is independent of the specific 1-fair policy used.

□

6.4 Trading Off Utility and Fairness

One straightforward way of trading off between the two objectives of fairness and principal's utility is to randomize between the two policies π_Γ^{TS} and π^*.

Definition 3 (OPT/TS-Mixing)**.** *The OPT/TS-Mixing ranking policy $\pi^{\mathrm{Mix},\phi}$ randomizes between π_Γ^{TS} and π_Γ^* with probabilities ϕ and $1-\phi$, respectively.*

The following lemma gives guarantees for such randomization (but we will later see that this strategy is suboptimal).

Lemma 6.4.1. *Consider two ranking policies π_1 and π_2 such that π_1 is ϕ_1-fair and π_2 is ϕ_2-fair. A policy that randomizes between π_1 and π_2 with probabilities q and $1-q$,* *respectively, is at least* $(q\phi_1 + (1-q)\phi_2)$*-fair and obtains expected utility* $qU(\pi_1 \,|\, \Gamma) + (1-q)U(\pi_2 \,|\, \Gamma)$.

Proof. Both the utility and fairness proofs are straightforward. The proof of fairness decomposes the probability of agent i being in position k under the mixing policy into the two constituent parts, then pulls terms through the sum. The proof of utility uses Equation (6.3) and linearity of expectations. We now give details of the proofs.

We write π_{Mix} for the policy that randomizes between π_1 and π_2 with probabilities q and $1-q$, respectively. Using Equation (6.3), we can express the utility of π_{Mix} as

$$\begin{aligned}
\mathrm{U}(\pi_{\text{Mix}} \,|\, \Gamma) &= \mathbb{E}_{\sigma \sim \pi_{\text{Mix}}, \boldsymbol{v} \sim \Gamma}\left[\mathrm{U}(\sigma \,|\, \boldsymbol{v})\right] \\
&= \mathbb{E}_{\boldsymbol{v} \sim \Gamma}\left[\sum_{\sigma} \pi_{\text{Mix}}(\sigma) \cdot \mathrm{U}(\sigma \,|\, \boldsymbol{v})\right] \\
&= \mathbb{E}_{\boldsymbol{v} \sim \Gamma}\left[\sum_{\sigma} (q \cdot \pi_1(\sigma) + (1-q) \cdot \pi_2(\sigma)) \cdot \mathrm{U}(\sigma \,|\, \boldsymbol{v})\right] \\
&= q \cdot \mathbb{E}_{\boldsymbol{v} \sim \Gamma}\left[\sum_{\sigma} \pi_1(\sigma) \cdot \mathrm{U}(\sigma \,|\, \boldsymbol{v})\right] + (1-q) \cdot \mathbb{E}_{\boldsymbol{v} \sim \Gamma}\left[\sum_{\sigma} \pi_2(\sigma) \cdot \mathrm{U}(\sigma \,|\, \boldsymbol{v})\right] \\
&= q\mathrm{U}(\pi_1 \,|\, \Gamma) + (1-q)\mathrm{U}(\pi_2 \,|\, \Gamma).
\end{aligned}$$

Similarly, we prove that π is at least $(q\phi_1 + (1-q)\phi_2)$-fair if π_1 and π_2 are ϕ_1- and ϕ_2-fair, respectively:

$$\begin{aligned}
\sum_{k'=1}^{k} p_{x,k'}^{(\pi)} &= \sum_{k'=1}^{k} q \cdot p_{x,k'}^{(\pi_1)} + (1-q) \cdot p_{x,k'}^{(\pi_2)} \\
&= q \cdot \sum_{k'=1}^{k} p_{x,k'}^{(\pi_1)} + (1-q) \cdot \sum_{k'=1}^{k} p_{x,k'}^{(\pi_2)} \\
&\geq q\phi_1 \cdot \mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}_{x,k}^{(\boldsymbol{v})} + (1-q)\phi_2 \cdot \mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}_{x,k}^{(\boldsymbol{v})}
\end{aligned}$$

$$= (q \cdot \phi_1 + (1-q) \cdot \phi_2) \cdot \mathbf{P}[\boldsymbol{v} \sim \Gamma] \mathcal{M}_{x,k}^{(\boldsymbol{v})},$$

where the inequality used that π_1 is ϕ_1-fair and π_2 is ϕ_2-fair. Hence, we have proved that π is $(q \cdot \phi_1 + (1-q) \cdot \phi_2)$-fair under Γ. □

Lemma 6.4.1 immediately implies:

Corollary 6.4.1. *The ranking policy $\pi^{\mathrm{Mix},\phi}$ is ϕ-fair.*

By definition, $\pi^{\mathrm{Mix},\phi=0}$ has the highest utility among all 0-fair ranking policies. Furthermore, all 1-fair policies achieve the same utility since the fairness axiom for $\phi = 1$ completely determines the marginal rank probabilities (Lemma 6.3.2). However, while $\pi^{\mathrm{Mix},\phi=0}$ and $\pi^{\mathrm{Mix},\phi=1}$ have the highest utility among 0-fair and 1-fair ranking policies, respectively, $\pi^{\mathrm{Mix},\phi}$ will typically not have maximum utility for the principal among all the ϕ-fair ranking policies for other values of $\phi \in (0,1)$. We illustrate this with a concrete example with $n = 3$ agents with the following example.

Example 2. Consider $n = 3$ agents, namely a, b, and c. Under Γ, their merits $v_a = 1$, $v_b \sim \text{Bernoulli}(1/2)$, and $v_c \sim \text{Bernoulli}(1/2)$ are drawn independently.[2] The position weights are $w_1 = 1, w_2 = 1$, and $w_3 = 0$.

Now, since $w_1 = w_2 = 1$ and agents b and c are i.i.d., any policy that always places agent a in positions 1 or 2 is optimal. In particular, this is true for the policy π^* which chooses uniformly at random from among $\sigma_1^* = \langle a, b, c \rangle$, $\sigma_2^* = \langle a, c, b \rangle$, $\sigma_3^* = \langle b, a, c \rangle$, and $\sigma_4^* = \langle c, a, b \rangle$.

[2] Technically, this distribution violates the assumption of non-identical merit of agents under Γ. This is easily remedied by adding — say — i.i.d. $\mathcal{N}(0, \epsilon)$ Gaussian noise to all v_i, with very small ϵ. We omit this detail since it is immaterial and would unnecessarily overload notation.

For the specific distribution Γ, assuming uniform random tie breaking, we can calculate the probabilities $\mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}_{x,k}^{(\boldsymbol{v})}$ in closed form:

$$\left(\mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}_{x,k}^{(\boldsymbol{v})}\right)_{x,k} = {}^{1}\!/_{24} \cdot \begin{pmatrix} 14 & 22 & 24 \\ 5 & 13 & 24 \\ 5 & 13 & 24 \end{pmatrix}.$$

The probability of a, b, c being *placed* in the top k positions by π^* can be calculated as follows:

$$\mathcal{P}^{(\pi^*)} = {}^{1}\!/_{24} \cdot \begin{pmatrix} 12 & 24 & 24 \\ 6 & 12 & 24 \\ 6 & 12 & 24 \end{pmatrix}.$$

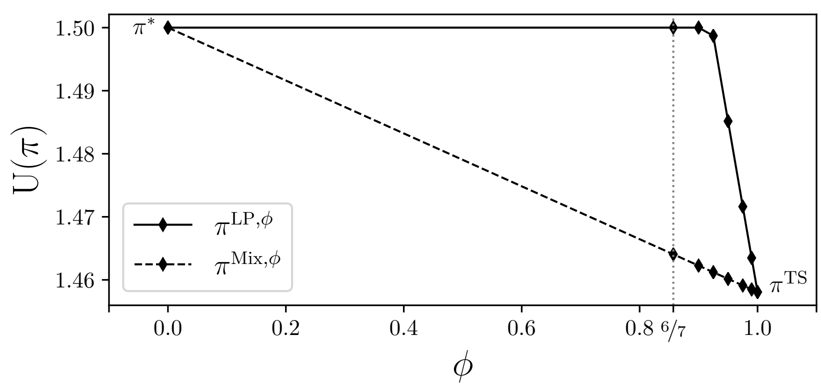


Figure 6.1: Utility of $\pi^{\mathrm{Mix},\phi}$ and $\pi^{\mathrm{LP},\phi}$ for Example 2 as one varies ϕ.

In particular, this implies that π^* is ϕ-fair for every $\phi \leq {}^{12}\!/_{14} = {}^{6}\!/_{7}$. This bound can be pushed up by slightly increasing the probability of ranking agent a at position 1 (hence increasing fairness to agent a in position 1 at the expense of agents b and c in positions 1–2). Figure 6.1 shows the principal's optimal utility for different fairness parameters ϕ, derived from the LP (6.4). This optimal utility is contrasted with the utility of $\pi^{\mathrm{Mix},\phi}$, which, by Lemma 6.4.1, is the convex combination of the utilities of π^* and π^{TS}.

6.4.1 Optimizing Utility for Approximately Fair Rankings

We now formulate a linear program for computing the policy $\pi^{\mathrm{LP},\phi}$ that maximizes the principal's utility, subject to being ϕ-fair. The variables of the linear program are the marginal rank probabilities $p^{(\pi)}_{x,k}$ of the distribution π to be determined. Then, by Equation (6.3) and linearity of expectation, the principal's expected utility can be written as $\mathrm{U}(\pi \mid \Gamma) = \sum_{x \in \mathcal{X}} \sum_k p^{(\pi)}_{x,k} \cdot \mathbb{E}_{\boldsymbol{v} \sim \Gamma}[v_x] \cdot w_k$.

We use this linear form of the utilities to write the optimization problem as the following LP with variables $p_{x,k}$ (omitting π from the notation):

$$
\begin{array}{lll}
\text{Maximize} & \sum_x \sum_k p_{x,k} \cdot \mathbb{E}_{\boldsymbol{v} \sim \Gamma}[v_x] \cdot w_k & \\
\text{subject to} & \sum_{k'=1}^{k} p_{x,k'} \geq \phi \cdot \mathbf{P}[\boldsymbol{v} \sim \Gamma] \mathcal{M}^{(\boldsymbol{v})}_{x,k} & \text{for all } x, k \\
& \sum_{k=1}^{n} p_{x,k} = 1 & \text{for all } x \\
& \sum_{i=1}^{n} p_{x,k} = 1 & \text{for all } k \\
& 0 \leq p_{x,k} \leq 1 & \text{for all } x, k.
\end{array}
\tag{6.4}
$$

In the LP, the first set of constraints captures ϕ-approximate fairness for all agents and positions, while the remaining constraints ensure that the marginal probabilities form a doubly stochastic matrix.

As a second step, the algorithm uses the Birkhoff-von Neumann (BvN) Decomposition of the matrix $\mathcal{P} = (p_{x,k})_{x,k}$ to explicitly obtain a distribution π over rankings such that π has marginals $p_{x,k}$. The Birkhoff-von Neumann Theorem (Birkhoff, 1940) states that the set of doubly stochastic matrices is the convex hull of the permutation matrices, which means that we can write $\mathcal{P} = \sum_\sigma q_\sigma \mathcal{P}^{(\sigma)}$, where $\mathcal{P}^{(\sigma)}$ is the binary permutation matrix corresponding to the deterministic ranking σ, and the q_σ form a probability distribution. It was already shown by

Birkhoff (1940) how to find a polynomially sparse decomposition in polynomial time.

In order to solve the Linear Program (6.4), one needs to know $\mathbf{P}[\boldsymbol{\nu} \sim \Gamma]\mathcal{M}^{(\boldsymbol{\nu})}_{x,k}$ for all i and k. For some distributions Γ (e.g., Example 2), these quantities can be calculated in closed form. For others, they can be estimated using Monte Carlo sampling. Such estimates may be off by small terms (which go to 0 as the number of samples grows large). This may result in a small loss of $\epsilon \to 0$ in the fairness guarantee which can be compensated by using a more aggressive fairness parameter $\phi' = \phi + \epsilon$ (when $\phi \leq 1 - \epsilon$). This translates the loss in fairness to a loss in utility, which also goes to 0 as the number of Monte Carlo samples grows.

6.5 Experimental Evaluation: MovieLens Dataset

To evaluate our approach in a recommendation setting with a realistic preference distribution, we design the following experimental setup based on the MovieLens 100K (ML-100K) dataset. The dataset contains 100,000 ratings, by 600 users, on 9,000 movies belonging to 18 genres (Harper and Konstan, 2015). In our setup, for each user, the principal is a recommender system that has to generate a ranking of movies $\mathcal{S}_g$ for one of the genres g (e.g., Horror, Romance, Comedy) according to a notion of *merit* of the movies we define as follows.

We define the (unknown) merit v_m of a movie m as the average rating of the movie across the user population[3] — this merit is unknown because most users have not seen/rated most movies. To be able to estimate this merit-based on ratings in the ML-100K dataset, and to concretely define its underlying distribution and the corresponding fairness criteria, we define a generative model of user ratings. The model assumes that each rating of a movie $m \in \mathcal{S}_g$ is drawn from a multinomial distribution over $\{1, 2, 3, 4, 5\}$ with (unknown) parameters $\boldsymbol{\theta}_m = (\theta_{m,1}, \ldots, \theta_{m,5})$.

Prior: These parameters themselves follow a Dirichlet prior $\boldsymbol{\theta}_m \sim \mathrm{Dir}(\boldsymbol{\alpha})$ with known parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)$. We assume that the parameters of the Dirichlet prior are of the form $\alpha_r = s \cdot p_r$ where s is a scaling factor and $p_r = \mathbb{P}[\text{Rating} = r \mid \mathcal{D}]$ denotes the marginal probability of observing the rating r in the full MovieLens dataset.

The scaling factor s determines the weight of the prior compared to the observed data since it acts as a pseudo-count in $\boldsymbol{\alpha}'$ below. For the sake of simplicity, we use $s = 1$ in the following for all movies and genres.

Posterior: Since the Dirichlet distribution is the conjugate prior of the multinomial distribution, the posterior distribution based on the ratings observed in the dataset $\mathcal{D}$ is also a Dirichlet distribution, but with parameters $\boldsymbol{\alpha}' = (\boldsymbol{\alpha} + \boldsymbol{N}_m) = (\alpha_1 + N_{m,1}, \ \ldots, \ \alpha_5 + N_{m,5})$ where $N_{m,r}$ is the number of ratings of r for the movie m in the dataset $\mathcal{D}$.

Utility Maximizing Ranking (π^*): We use the DCG function (Burges et al., 2005) with position weights $w_k = {}^{1}/_{\log_2(1+k)}$ as our utility measure. Since the weights are

[3] For a personalized ranking application, an alternative would be to choose each user's (mostly unknown) rating as the merit criterion instead of the average rating across the user population.

indeed strictly decreasing, as described in Section § 6.3, the optimal ranking policy π^* sorts the movies (for the particular query) by decreasing expected merit, which is the expected average rating $\overline{v}_m$ under the posterior Dirichlet distribution, and can be computed in closed form as follows:

$$\overline{v}_m \triangleq \mathbb{E}_{\boldsymbol{\theta}\sim\mathbb{P}[\boldsymbol{\theta}_m \mid \mathcal{D}]} \left[v_m(\boldsymbol{\theta})\right] \;=\; \sum_{r=1}^{5} r \cdot \frac{\alpha_r + N_{m,r}}{\sum_{r'} \alpha_{r'} + N_{m,r'}}. \tag{6.5}$$

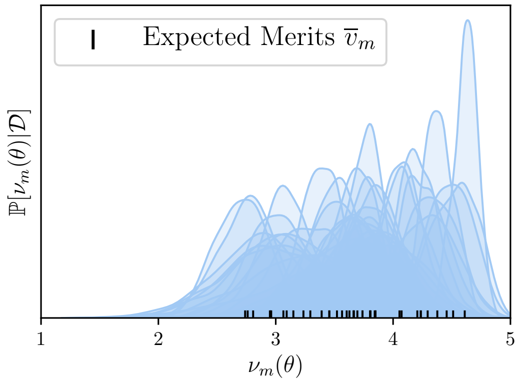
Fully Fair Ranking Policy (π^{TS}): A fair ranking in this case ensures that, for all positions k, a movie is placed in the top k positions according to the posterior merit distribution. In this setup, a fully fair ranking policy π^{TS} is obtained by sampling the multinomial parameters $\boldsymbol{\theta}_m$ for each movie $m \in \mathcal{S}_g$ and computing $v_m(\boldsymbol{\theta}_m)$ to rank them:

$$\pi^{\mathrm{TS}}(\mathcal{S}_g) \sim \operatorname*{argsort}_m v_m(\boldsymbol{\theta}_m) \text{ s.t. } \boldsymbol{\theta}_m \sim \mathbf{P}\boldsymbol{\theta}_m|\mathcal{D}.$$

LP Ranking Policy ($\pi^{\mathrm{LP},\phi}$): The ϕ-fair policies $\pi^{\mathrm{LP},\phi}$ require the principal to have access to the probabilities $\mathbf{P}[\boldsymbol{v} \sim \Gamma]\mathcal{M}^{(\boldsymbol{v})}_{m,k}$ which we estimate using $5 \cdot 10^4$ Monte Carlo samples, so that any estimation error becomes negligible.

OPT/TS-Mixing Ranking Policy ($\pi^{\mathrm{Mix},\phi}$): The policies $\pi^{\mathrm{Mix},\phi}$ randomize between the fully fair and utility-maximizing ranking policies with probabilities ϕ and $1 - \phi$, respectively.

Observations and Results. In the experiments presented, we used the ranking policies π^*, π^{TS}, $\pi^{\mathrm{Mix},\phi}$ and $\pi^{\mathrm{LP},\phi}$ to create separate rankings for each of the 18 genres. For each genre, the task is to rank a random subset of 40 movies from that genre. To get a posterior with an interesting degree of uncertainty, we take a 10% i.i.d. samples from $\mathcal{D}$ to infer the posterior for each movie. We observe that the results are qualitatively consistent across genres, and we thus focus



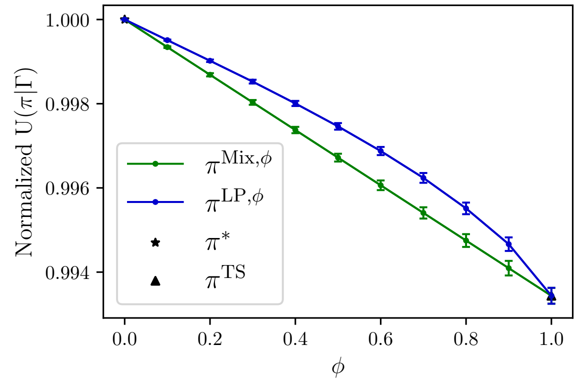


Figure 6.2: (a) Posterior distribution of ratings (merit) for a subset of "Comedy" movies, (b) Tradeoff between Utility and Fairness, as captured by ϕ.

on detailed results for the genre "Comedy" as a representative example. Its posterior merit distribution over a subset is visualized in Figure 6.2(a). Note that substantial overlap exists between the marginal merit distributions of the movies, indicating that as opposed to π^* (which sorts based on the expected merits), the policy π^{TS} will randomize over many different rankings.

Observation 1: We evaluate the cost of fairness to the principal in terms of loss in utility, as well as the ability of $\pi^{\mathrm{LP},\phi}$ to minimize this cost for ϕ-fair rankings. Figure 6.2(b) shows this cost in terms of expected Normalized DCG (i.e., NDCG = ${}^{\mathrm{DCG}}/_{\max(\mathrm{DCG})}$ as in (Järvelin and Kekäläinen, 2002)). These results are averaged over 20 runs with different subsets of movies and different training samples. The leftmost end corresponds to the NDCG of π^*, while the rightmost point corresponds to the NDCG of the 1-fair policy π^{TS}.

The drop in NDCG is below one percentage point, which is consistent with the results for the other genres. We also conducted experiments with other values of s, data set sizes, and choices of w_k; even under the most extreme conditions, the drop was at most 2 percent. While this rather small drop may be surprising at first, we point out that uncertainty in the estimates affects the utility of both π^* and π^{TS}. By industry standards, a 2% drop in NDCG is considered quite substantial; however, it is not catastrophic and hence bodes well for possible adoption.

Observation 2: Figure 6.2(b) also compares the tradeoff in NDCG in response to the fairness approximation parameter ϕ for both $\pi^{\text{Mix},\phi}$ and $\pi^{\text{LP},\phi}$. We observe that the utility-optimal policy $\pi^{\text{LP},\phi}$ provides gains over $\pi^{\text{Mix},\phi}$, especially for large values of ϕ. Thus, using $\pi^{\text{LP},\phi}$ can further reduce the cost of fairness discussed above.

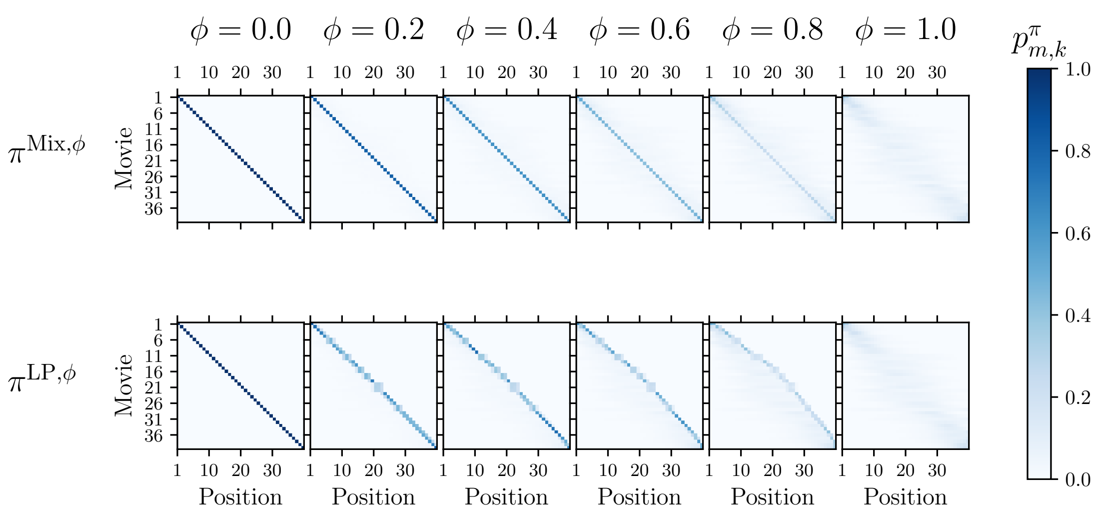


Figure 6.3: Comparison of marginal rank distribution matrices for $\pi^{\text{Mix},\phi}$ and $\pi^{\text{LP},\phi}$ on "Comedy" movies.

Observation 3: To provide intuition about the difference between $\pi^{\text{Mix},\phi}$ and $\pi^{\text{LP},\phi}$, Figure 6.3 visualizes the marginal rank distributions $p_{m,k}$, i.e., the probability that movie m is ranked at position k. The key distinction is that, for intermediate values of ϕ, $\pi^{\mathrm{LP},\phi}$ exploits a non-linear structure in the ranking distribution (to achieve a better trade-off) while $\pi^{\mathrm{Mix},\phi}$ merely interpolates linearly between the solutions for $\phi = 0$ and $\phi = 1$.

Based on these observations, in general, the utility loss due to fairness is small, and can be further reduced by optimizing the ranking distribution with the LP-based approach.These results are based on the definition of merit as the average rating of movies over the entire user population. A more realistic setting would personalize rankings for each user, with merit defined as the expected relevance of a movie to the user. In our experiments, the under such a setup were quite similar, and are hence omitted for brevity and clearer illustration.

6.6 Real-World Experiment: Paper Recommendation

To study the effect of deploying a fair ranking policy in a real ranking system, we built and fielded a paper recommendation system at a scientific conference. The goal of the experiment is to understand the impact of fairness under real user behavior, as opposed to simulated user behavior that is subject to modeling assumptions. Specifically, we seek to answer two questions: (a) Does a fair ranking policy lead to a more equitable distribution of exposure among the papers? (b) Does fairness substantially reduce the utility of the system to the users?

The users of the paper recommendation system were the participants of a virtual conference held in 2020. Signup and usage of the system was voluntary. Each user was recommended a personalized ranking of the papers published at the conference. This ranking was produced either by σ^* or by π^{TS}, and the assignment of users to treatment (π^{TS}) or control (σ^*) was randomized.

Modeling the Merit Distribution. The merit of a paper for a particular user is based on a relevance score $\mu_{u,i}$ that relates features of the user (e.g., bag-of-words representation of recent publications, co-authorship) to features of each conference paper (e.g., bag-of-words representation of paper, citations). Most prominently, the relevance score $\mu_{u,i}$ contains the TFIDF-weighted cosine similarity between the bag-of-words representations.

We model the uncertainty in $\mu_{u,i}$ with regard to the true relevance as follows. First, we observe that all papers were accepted to the conference and thus must have been deemed relevant to at least some fraction of the audience by the peer reviewers. This implies that papers with uniformly low $\mu_{u,i}$ across all/most participants are not irrelevant; we merely have high uncertainty as to which participants the papers are relevant to. For example, papers introducing new research directions or bringing in novel techniques may have uniformly low scores $\mu_{u,i}$ under the bag-of-words model that is less certain about who wants to read these papers compared to papers in established areas. To formalize uncertainty, we make the assumption that a paper's relevance to a user follows a normal distribution centered at $\mu_{u,i}$, and with standard deviation equal to δ_i (dependent only on the paper, not the user) such that $\max_u \mu_{u,i} + \gamma \cdot \delta_i = 1 + \epsilon$. (For our experiments, we chose $\epsilon = 0.1$ and $\gamma = 2$.) This choice of δ_i ensures that there exists at least one user u such that the (sampled) relevance score $\hat{\mu}_{u,i}$ is greater than 1 with some significant probability; more specifically, we ensure that the probability of having relevance $1 + \epsilon$ is at least as large as that of exceeding the mean by two standard deviations. Furthermore, $\epsilon > 0$ ensures that all papers have a

	Number of Users with activity		Average Activity Per User	
	π^*	π^{TS}	π^*	π^{TS}
Total Number of users	213	248	-	-
Num. of pages examined	-	-	10.8075	10.7984
`Read Abstract`	92	101	3.7230	2.6774
`Add to Calendar`	51	50	1.4366	0.8508
`Read PDF`	40	52	0.5258	0.5323
`Add Bookmark`	16	13	0.3192	0.6129

Table 6.1: User engagement with the Paper Recommendation System under the two conditions π^* and π^{TS}. None of the differences are statistically significant.

non-deterministic relevance distribution, even papers with $\max_u \mu_{u,i} = 1$.

Ranking Policies. Users in the control group $\mathcal{U}_{\pi^*}$ received rankings in decreasing order of $\mu_{u,i}$. Users in the treatment group $\mathcal{U}_{\pi^{\text{TS}}}$ received rankings from the fair policy that sampled scores from the uncertainty distribution, $\hat{\mu}_{u,i} \sim \mathcal{N}(\mu_{u,i}, \delta_i)$, and ranked the papers by decreasing $\hat{\mu}_{u,i}$.

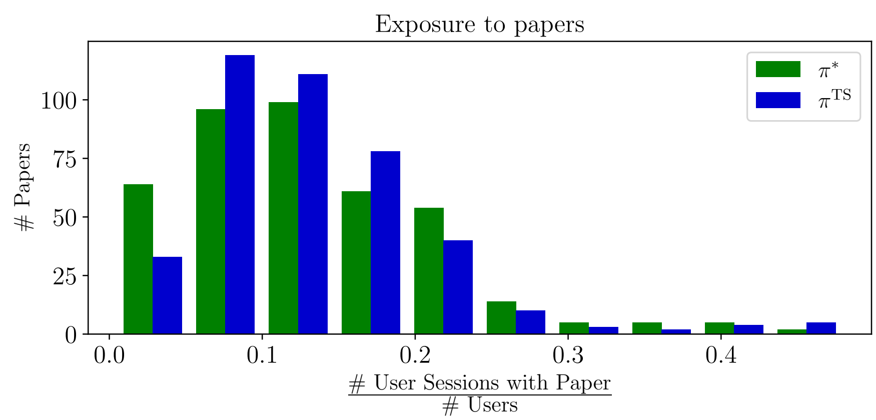


Figure 6.4: Distribution of the exposure of papers in treatment and control. Exposure is defined as the average number of sessions per user in which the paper was shown to the user.

Results and Observations. We first analyze if the fair policy provided more equitable exposure to the papers. In this real-world evaluation, exposure is not equivalent to rank, but depends on whether users actually browsed to a given position in the ranking. Users could browse their ranking in pages of 5 recommendations each; we count a paper as *exposed* if the user scrolled to its page.

Observation 1: Figure 6.4 compares the histogram of exposure of the papers in the treatment and control groups. Under the fair policy, the number of papers in the lowest tier of exposure is roughly halved compared to the control condition. This verifies that the fair policy does have an impact on exposure in a real-world setting, and it aligns with our motivation that a fair ranking policy distributes exposure more equally among the ranked agents. This is also supported by comparing the Gini inequality coefficient (Gini, 1936) for the two distributions: $G(\pi^*) = 0.3302$, while $G(\pi^{\mathrm{TS}}) = 0.2996$ (where a smaller coefficient means less inequality in the distribution).

Observation 2: To evaluate the impact of fairness on user engagement, we analyze a range of engagement metrics as summarized in Table 6.1. While a total of 923 users signed up for the system ahead of the conference (and were randomized into treatment and control groups), 462 never came to the system after all. Out of the users that came, 213 users were in $\mathcal{U}_{\pi^*}$, and 248 users were in $\mathcal{U}_{\pi^{\mathrm{TS}}}$. Note that this difference is not caused by the treatment assignment since users had no information about their assignment/ranking before entering the system. The first engagement metric we computed is the average number of pages that users viewed under both conditions. With roughly 10.8 pages (about 54 papers), engagement under both conditions was almost identical. Users also had other options to engage, but there is no clear difference between the conditions, either. On average, they read more paper abstracts and added more papers to their calendar under the control condition, but read more PDF and added more bookmarks under the treatment condition. However, none of these differences is significant at the 95% level for either a Mann-Whitney U test or a two-sample t-test. While the sample size is small, these findings align with the findings on the synthetic data, namely that fairness did not appear to place a significantly large cost on the principal (here representing the users).

6.7 Summary and Discussion

We believe that the focus on uncertainty presented in this chapter constitutes a principled approach to capturing the intuitive notion of fairness to agents with similar features – rather than focusing on the features themselves. A key insight is that the features' similarity entails significant statistical uncertainty about which agent has more merit, and randomization for ranking provides a way to fight fire with fire and axiomatize fairness in the presence of such uncertainty.

Our work raises a wealth of questions for future work. Perhaps most importantly, as discussed in Section § 6.2.4, to operationalize our proposed notion of fairness, it is essential to derive principled merit distributions Γ based on the observed features. Our experiments were based on "reasonable" notions of merit distributions and concluded that fairness might not have to be very expensive to achieve for a principal. However, more experimental work is needed to truly evaluate the impact of fair policies on the achieved utility. It would be particularly intriguing to investigate which types of real-world settings lend themselves to implementing fairness at little cost and which force a steeper tradeoff between the two objectives.

Our work also raises several interesting theoretical questions. In Section §6.2.4, we show one setting in which forcing the principal to use a fair policy drastically increases the principal's incentives to form a more accurate posterior Γ. We ask: will the incentives of a principal to learn a better posterior Γ always (weakly) increase if the principal is forced to be fairer? If it is indeed true, this would provide a fascinating additional benefit of fairness requirements.

Another interesting question concerns the utility loss incurred by using the policy OPT/TS-Mixing. As we saw in Section §6.4, OPT/TS-Mixing is in general not optimal. However, in the example from Section §6.4 as well as in our experiments in Section §6.5, the loss in utility was quite small. An interesting question would be to bound the worst-case loss in the utility of OPT/TS-Mixing, compared to the LP-based policy. In particular, this question is of interest due to the simplicity of the OPT/TS-Mixing policy; it does not require the computationally expensive solution of an LP or an explicit estimate of marginal rank probabilities under Γ.

CHAPTER 7

CONCLUSION AND OUTLOOK

Despite the growing influence of online information systems on our society and economy, research on studying the impact of rankings and underlying algorithms has been relatively underexplored. Learning-to-Rank algorithms have provided tremendous gains in terms of conventional evaluation metrics that measure the usefulness of these systems for the users. However, we identified that the mechanism of exposure allocation has an impact on the fairness and dynamics of the system, and an algorithm that only optimizes for utility may lead to rankings that are unfair and even undesirable (Chapter 1).

This thesis proposed a framework for formulating fairness constraints on rankings using its expressive power to link merit, relevance, exposure, and impact. We started by formalizing a post-processing method to ensure ranking fairness while optimizing user utility for the case when the relevance of each item was already given (Chapter 3). A key feature of this framework was introducing stochastic rankings that facilitate both the optimization task and the feasibility of the fairness constraint.

We then addressed the drawback of using the post-processing method on estimated relevances by proposing an end-to-end LTR algorithm called FAIR-PG-RANK that can directly optimize any information retrieval utility metric while satisfying a wide range of fairness criteria (Chapter 4). The experiments on two datasets demonstrated that FAIR-PG-RANK generalizes to unseen queries and also learns to ignore biased features.

In a more realistic setting, where the learning-to-rank algorithm dynamically adapts as more user-feedback data becomes available, we recognized two intertwined challenges — first, the user feedback is biased due to the positions of documents in the presented rankings, and second, the disparity in exposure between two groups or individuals may promote a rich-get-richer dynamic if not handled. In Chapter 5, these two challenges are addressed simultaneously by a P-controller based ranking algorithm that uses unbiased estimators for utility and fairness, dynamically adapting both as more data becomes available. The algorithm exhibits favorable theoretical convergence properties as well as is highly practical and robust when employed in an experimental setup based on a real-world user-feedback dataset.

In Chapter 6, we considered fairness for rankings under uncertainty in the knowledge of merit of the items. We formalized an axiomatic framework that follows the principle that the relative ordering between a pair of items under a fair ranking policy must be consistent with the existing evidence about their relative ordering. This principle is formalized into a computational framework that leads to an algorithm to compute (approximately) fair rankings that are also optimal with respect to expected utility to the user.

While the methods proposed in this dissertation are just beginning to answer the question of building fair and equitable multistakeholder systems, they represent real progress by contributing a conceptual framework that allows product stakeholders to express application-specific fairness goals. These advances in formulating notions of fairness for rankings have led to keen interest from the industry in rethinking production ranking algorithms. Some notable examples include LinkedIn Talent search (Geyik et al., 2019) and connection recommendations (Nandy et al., 2021), a large-scale production recommender system at Google (Beutel et al., 2019), and Spotify Music Recommendations (Mehrotra et al., 2018).

7.1 Future Directions

As machine learning-based rankings become an indispensable part of any online platform, they have a direct and an indirect impact on our future selves, our society as well as the economy. With this growing influence, it is valuable to formalize the study of the impacts of such systems from the perspective of different stakeholders. This section lays out some ongoing and relevant research directions for the future.

Diversity and Fairness in Rankings

Fairness of exposure in rankings is seemingly close to the idea of diversity in information retrieval. While both these goals lead to rankings that do not follow the probability ranking principle (Robertson, 1977), the two are fundamentally different. In particular, algorithms that optimize diversity merely use a different model of user utility while still optimizing exclusively for user utility without any consideration of the ranked items. One difference between conventional and diversified utility measures is that they are not necessarily modular (i.e., linearly additive) in the set of ranked items, but that they can model dependencies between the items – most commonly in the form of a submodular set function (Yue and Guestrin, 2011). So while diversity and fairness of exposure have different goals, they appear to have mutually compatible effects on the rankings they produce. An interesting future research problem is to study diversified ranking algorithms with provable fairness guarantees. One way to achieve this is by defining submodular utility measures for evaluation and optimization (e.g., (Diaz et al., 2020)). However, optimizing such metrics is substantially more challenging than the modular utility metrics (e.g., DCG) considered so far, but existing connections between submodular set functions and linear programming (Calinescu et al., 2007) may provide a path to tackling the problem with the techniques developed in Chapter 3.

Position Bias, Trust Bias, and User Behavior

Past studies have provided evidence that position in the ranking not only affects exposure (through position bias) but also the user's valuation of the item (i.e., the user's perception of relevance) (Joachims et al., 2007). This is commonly referred to as *trust bias*, where the user takes the rank of the item as evidence of its relevance. Trust bias is often intertwined with the position-based attention bias when we observe clicks. However, trust bias is also fundamentally different since its causal pathway is through the user's valuation of the item and not through the missingness in the data as is the case with position bias, hence, making it an interesting causal inference challenge to solve either through minor interventions or by harnessing natural experiments in observational data.

While correcting for position and trust bias may result in estimates that are correct on average, the variance of these estimates may still be substantially different between groups. This variance problem is well known for IPS estimators used for counterfactual evaluation and learning (Swaminathan and Joachims, 2015). Moreover, the estimates are likely to have a higher variance for historically underrepresented items because of the lack of data. This means that a single point estimate can be far off, leading to potential unfairness. To overcome this problem, it is important to study how active exploration can be introduced into the estimation problem of (Morik et al., 2020), leading to the largely underexplored questions of fairness in online learning (Patil et al., 2020; Joseph et al., 2016). In this way, active exploration adds another component to both the trade-off between short-term and long-term user utility and the system's fairness properties.

Studying these effects also requires a scientific inquiry into user behavior models. Since the work on exposure-based item fairness and studying the impact on users uses certain assumptions about the user behavior, it becomes very important to examine and validate these behavioral models. Such a study would open the door to making definitive claims about the effect of these systems on their users. In an industry setting, there is a huge scope in understanding this better via minor interventions (Joachims et al., 2017), or harnessing natural experiments in observational data (Sharma et al., 2015).

Fair and Safe Exploration in Sequential Recommendations

Given the considerable success of Reinforcement Learning (RL) in games, robotics, and physical system control, it has also become a common framework to train recommender systems that optimize user feedback over the entire sequence (Chen et al., 2019). However, the use of RL for recommendations brings new challenges of its own, for example, using off-policy logged data to optimize the recommendation policy, exploration in the online setting, and the enormous size of the action space. From the perspective of fairness and safety, exploration, in particular, raises some complex and unanswered ethical questions. The hope is that, even though exploration temporarily degrades the user experience, it leads to improvements in the long run. On the other hand, there is a high chance that some groups of users share much of the burden of exploration without sufficient payoff in the longer term. The online learning-to-rank algorithm presented in Chapter 5 only exploits user-driven exploration and hence a proper exploration technique could ensure faster convergence rates as well as fairness guarantees to items and users. This is a fascinating topic of study both from the ethical and algorithmic standpoint, and it aligns well with the motivations of this thesis.

Dynamic Effects on Items.

Fairness is only one reason for directly controlling how exposure is allocated based on merit. As illustrated in the examples in Chapter 1, exposure allocation also has implications on how items may participate in the platform. In order to quantify these effects, one can explore decisions-theoretic, and game-theoretic models of how different exposure allocation schemes influence the behavior of the items (Tennenholtz and Kurland, 2019). The following are two examples.

First, an important game-theoretic problem is to show whether an alternate exposure-allocation scheme exists under which manipulative Search Engine Optimization (SEO) is no longer profitable. Consider, for example, an exposure-allocation scheme where exposure is proportional to relevance as in Equation (3.7). Under such a scheme, any investment by an item on illicitly improving its relevance estimates (such as through SEO) by ϵ would only be rewarded with a proportional gain in exposure, and not with a highly disproportionate gain like in a conventional deterministic ranking policy. An interesting problem here is to identify such exposure-allocation schemes that maintain high utility to the users but where manipulative SEO is no longer profitable under plausible economic assumptions.

Second, exposure allocation has dynamic implications on the participation of items in the platform. Similar to the introductory example in Section § 1.1.1, a music streaming service may want to foster a large community of contributing artists (DiCola, 2013). Through flattening the exposure allocation scheme of its recommender system, the service can control that the streaming revenue is not only concentrated on a few superstar artists but that a substantial number of artists can make a living. This might avoid monopolization, contributing a broader inventory of music and keeping prices low for the consumers. However, making the exposure allocation scheme too flat may result in spreading the streaming revenue too thin so that again fewer artists can make a living, which again leads to decreased participation in the platform. A family of economic models may help product designers and decision-makers analyze these effects enabling them to make informed decisions about which exposure allocation scheme to use.

Dynamic Effects on Users. A related future direction is to study how exposure allocation affects the users in an information ecosystem over time. Platforms such as social media, online marketplaces, and news recommenders are designed to maximize their profits, and most often, maximizing profit is analogous to optimizing user engagement that comes at the cost of manipulating user attention. Consider two scenarios again – first, consider an application such as image search (Section § 1.1.2) where users expect to see a ranking that is reflective of an underlying relevance distribution, and second, consider the top-news homepage ranking from Section §1.1.1. In the former scenario, Kay et al. (2015); Otterbacher et al. (2017) found that gender biases in image search results affect user's beliefs about different occupations, which is a real-world example of how biased information systems affect user's perceptions. In the latter example from Chapter 1, a minority-view user may feel alienated by a ranking that represents the relevances of the majority. This user is then less likely to return to the platform, leaving behind an "echo chamber" of users with similar preferences (Levy, 2021; Pariser, 2011). Since managing a broad and diverse user base is essential for these platforms to maintain their competitiveness, research on studying such dynamic effects on individual users and the user population becomes crucial.

BIBLIOGRAPHY

Abdollahpouri, H., Adomavicius, G., Burke, R., Guy, I., Jannach, D., Kamishima, T., Krasnodebski, J., and Pizzato, L. (2019). Beyond personalization: Research directions in multistakeholder recommendation. *arXiv preprint arXiv:1905.01986*. [Pages 28 and 90.]

Abdollahpouri, H., Burke, R., and Mobasher, B. (2017). Controlling popularity bias in learning-to-rank recommendation. In *RecSys*. [Page 90.]

Adamic, L. A. and Huberman, B. A. (2000). Power-law distribution of the world wide web. *Science*. [Pages 85 and 90.]

Agarwal, A., Takatsu, K., Zaitsev, I., and Joachims, T. (2019a). A general framework for counterfactual learning-to-rank. In *SIGIR*. [Page 86.]

Agarwal, A., Zaitsev, I., Wang, X., Li, C., Najork, M., and Joachims, T. (2019b). Estimating position bias without intrusive interventions. In *WSDM*. [Pages 23, 43, 64, 94, and 98.]

Ai, Q., Bi, K., Luo, C., Guo, J., and Croft, W. B. (2018). Unbiased learning to rank with unbiased propensity estimation. In *SIGIR*. [Page 91.]

Ali, M., Sapiezynski, P., Bogen, M., Korolova, A., Mislove, A., and Rieke, A. (2019). Discrimination through optimization: How facebook's ad delivery can lead to biased outcomes. *CSCW*. [Page 32.]

Asudeh, A., Jagadish, H., Stoyanovichz, J., and Das, G. (2019). Designing fair ranking schemes. *ICDM*. [Page 26.]

Baeza-Yates, R. (2018). Bias on the web. *Communications of the ACM*. [Pages 31 and 90.]

Barocas, S., Hardt, M., and Narayanan, A. (2019). *Fairness and Machine Learning*. fairmlbook.org. `http://www.fairmlbook.org`. [Page 25.]

Barocas, S. and Selbst, A. D. (2016). Big data's disparate impact. *Cal. L. Rev.* [Page 24.]

Beam, M. A. (2014). Automating the news: How personalized news recommender system design choices impact news reception. *Communication Research*. [Page 90.]

Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*. [Page 15.]

Bequette, B. W. (2003). *Process control: modeling, design, and simulation*. Prentice Hall Professional. [Page 102.]

Beutel, A., Chen, J., Doshi, T., Qian, H., Wei, L., Wu, Y., Heldt, L., Zhao, Z., Hong, L., Chi, E. H., and Goodrow, C. (2019). Fairness in recommendation ranking through pairwise comparisons. In *KDD*. [Pages 27, 28, 34, 130, and 150.]

Biega, A., Diaz, F., Ekstrand, M., and Kohlmeier, S. (2019). Trec 2019 fair ranking track. [Page 88.]

Biega, A. J., Gummadi, K. P., and Weikum, G. (2018). Equity of attention: Amortizing individual fairness in rankings. *SIGIR*. [Pages 26, 53, 54, 59, 65, 86, 88, 96, 119, and 130.]

Birkhoff, G. (1940). *Lattice theory*, volume 25. American Mathematical Soc. [Pages 41, 53, 120, 125, 136, and 137.]

Brams, S. and Taylor, A. (1996). *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press. [Page 29.]

Brams, S. J. and Taylor, A. D. (1995). An envy-free cake division protocol. *The American Mathematical Monthly*. [Page 29.]

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *ICML*, pages 89–96. ACM. [Pages 20, 69, and 138.]

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. *Learning*. [Pages 20 and 91.]

Burke, R., Sonboli, N., and Ordonez-Gauger, A. (2018). Balanced neighborhoods for multi-sided fairness in recommendation. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*. [Page 28.]

Butterly, A. (2015). Google image search for ceo has barbie as first female result. [Page 7.]

Calders, T., Kamiran, F., and Pechenizkiy, M. (2009). Building classifiers with independency constraints. In *Data mining workshops, ICDMW*, pages 13–18. [Pages 25, 26, and 46.]

Calinescu, G., Chekuri, C., Pal, M., and Vondrak, J. (2007). Maximizing a submodular set function subject to a matroid constraint. In *Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 182–196. [Page 151.]

Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., and Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In *ICML*, pages 129–136. ACM. [Pages 20 and 69.]

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336. [Page 30.]

Carterette, B. (2011). An analysis of np-completeness in novelty and diversity ranking. *Information Retrieval*, 14(1):89–106. [Page 30.]

Carterette, B. and Jones, R. (2007). Evaluating search engines by modeling the relationship between relevance and clicks. *Computer Science Department Faculty Publication Series*, page 26. [Page 21.]

Celis, L. E., Straszak, D., and Vishnoi, N. K. (2018). Ranking with fairness constraints. *ICALP*. [Pages 26, 54, and 130.]

Celma, Ò. and Cano, P. (2008). From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, page 5. ACM. [Page 28.]

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press. [Page 85.]

Chandler, J. A. (2006). A right to reach an audience: An approach to intermediary bias on the internet. *Hofstra L. Rev.*, 35:1095. [Page 33.]

Chang, C.-S., Chen, W.-J., and Huang, H.-Y. (1999). On service guarantees for input-buffered crossbar switches: a capacity decomposition approach by birkhoff and von neumann. In *IWQoS*, pages 79–86. IEEE. [Page 41.]

Chapelle, O. and Chang, Y. (2011). Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*, pages 1–24. [Pages 62, 75, 78, and 79.]

Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., and Chi, E. H. (2019). Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 456–464. [Page 152.]

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163. [Page 25.]

Chuklin, A., Markov, I., and Rijke, M. d. (2015). Click models for web search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7. [Pages 3, 23, 91, 93, 94, and 106.]

Ciampaglia, G. L., Nematzadeh, A., Menczer, F., and Flammini, A. (2018). How algorithmic popularity bias hinders or promotes quality. *Scientific reports*. [Page 90.]

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *SIGIR*, pages 659–666. [Page 31.]

Cossock, D. and Zhang, T. (2006). Subset ranking using regression. In *COLT*. [Page 19.]

Crammer, K. and Singer, Y. (2002). Pranking with ranking. In *NeurIPS*. Neural information processing systems foundation. [Page 20.]

Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. (2008). An experimental comparison of click position-bias models. In *WSDM*. [Pages 3, 22, 50, 94, and 98.]

Datta, A., Tschantz, M. C., and Datta, A. (2015). Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112. [Page 32.]

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository. [Pages 74, 75, and 83.]

Diaz, F., Mitra, B., Ekstrand, M. D., Biega, A. J., and Carterette, B. (2020). Evaluating stochastic rankings with expected exposure. In *CIKM*. [Pages 26 and 151.]

DiCola, P. (2013). Money from music: Survey evidence on musicians' revenue and lessons about copyright incentives. *Ariz. L. Rev.*, 55:301. [Page 154.]

Dufossé, F. and Uçar, B. (2016). Notes on birkhoff–von neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 497:108–115. [Page 41.]

Dupret, G. E. and Piwowarski, B. (2008). A user browsing model to predict search engine click data from past observations. In *SIGIR*, pages 331–338. [Page 23.]

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *ITCS*, pages 214–226. [Pages 24, 25, 47, 125, and 128.]

Dwork, C., Kim, M. P., Reingold, O., Rothblum, G. N., and Yona, G. (2019). Learning from outcomes: Evidence-based rankings. In *FOCS*. [Pages 25, 27, and 130.]

Edelman, B., Luca, M., and Svirsky, D. (2017). Racial discrimination in the sharing economy: Evidence from a field experiment. *American Economic Journal: Applied Economics*, 9(2):1–22. [Page 32.]

Edelman, B. G. and Luca, M. (2014). Digital discrimination: The case of airbnb. com. *Harvard Business School NOM Unit Working Paper*. [Page 32.]

Ekstrand, M. D., Tian, M., Kazi, M. R. I., Mehrpouyan, H., and Kluver, D. (2018). Exploring author gender in book rating and recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 242–250. ACM. [Page 28.]

Ensign, D., Friedler, S. A., Neville, S., Scheidegger, C., and Venkatasubramanian, S. (2018). Runaway feedback loops in predictive policing. In *Conference of Fairness, Accountability, and Transparency*. [Page 90.]

Fan, H. and Poole, M. S. (2006). What is personalization? perspectives on the design and implementation of personalization in information systems. *Journal of Organizational Computing and Electronic Commerce*, 16(3-4):179–202. [Page 15.]

Fang, Z., Agarwal, A., and Joachims, T. (2019). Intervention harvesting for context-dependent examination-bias estimation. In *SIGIR*. [Pages 23, 64, and 98.]

Ferrara, E. and Yang, Z. (2015). Measuring emotional contagion in social media. *PloS one*, 10(11):e0142390. [Page 31.]

Ferraro, A., Serra, X., and Bauer, C. (2021). Break the loop: Gender imbalance in music recommenders. In *Proceedings of the 2021 Conference on Human Information Interaction and Retrieval*, pages 249–254. [Page 2.]

Fleder, D. and Hosanagar, K. (2009). Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science*, 55(5):697–712. [Page 28.]

Fox, S., Karnawat, K., Mydland, M., Dumais, S., and White, T. (2005). Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168. [Page 21.]

Germano, F., Gómez, V., and Mens, G. L. (2019). The few-get-richer: a surprising consequence of popularity-based rankings. *WWW*. [Page 90.]

Geyik, S. C., Ambler, S., and Kenthapadi, K. (2019). Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. *KDD*. [Pages 34 and 149.]

Gillespie, T. (2014). The relevance of algorithms. *Media technologies: Essays on communication, materiality, and society*, 167(2014):167. [Page 33.]

Gini, C. (1936). On the measure of concentration with special reference to income and statistics. *Colorado College Publication, General Series*. [Page 145.]

Granka, L. A. (2010). The politics of search: A decade retrospective. *The Information Society*, 26(5):364–374. [Page 8.]

Grimmelmann, J. (2011). Some skepticism about search neutrality. *The Next Digital Decade: Essays on the future of the Internet*, page 435. [Pages 2, 7, 8, and 33.]

Guo, F., Liu, C., and Wang, Y. M. (2009). Efficient multiple-click models in web search. In *WSDM*. ACM. [Page 23.]

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. In *NIPS*, pages 3315–3323. [Page 25.]

Harper, F. M. and Konstan, J. A. (2015). The movielens datasets: History and context. *TIIS*. [Pages 112 and 137.]

Herbrich, R. (1999). Support vector learning for ordinal regression. In *NIPS*, pages 97–102. [Page 20.]

Herbrich, R., Graepel, T., and Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers*, 88(2):115–132. [Page 90.]

Hofmann, K., Whiteson, S., and de Rijke, M. (2013). Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*. [Pages 91 and 100.]

Horvitz, D. G. and Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*. [Page 99.]

Imbens, G. W. and Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press. [Page 99.]

Introna, L. D. and Nissenbaum, H. (2000). Shaping the web: Why the politics of search engines matters. *The information society*, 16(3):169–185. [Page 8.]

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446. [Pages 37, 94, 127, and 140.]

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *KDD*, pages 133–142. [Pages 20, 21, and 91.]

Joachims, T. (2006). Training linear svms in linear time. In *KDD*, pages 217–226. ACM. [Pages 20, 78, and 79.]

Joachims, T., Finley, T., and Yu, C.-N. J. (2009). Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59. [Page 69.]

Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., and Gay, G. (2007). Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM TOIS*. [Pages 23, 64, 85, 86, 87, 93, and 151.]

Joachims, T., Granka, L. A., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *SIGIR*. [Pages 3, 21, 22, 33, and 50.]

Joachims, T., Swaminathan, A., and Schnabel, T. (2017). Unbiased learning-to-rank with biased feedback. In *WSDM*. [Pages 21, 23, 43, 55, 64, 70, 85, 86, 87, 91, 94, 98, 99, and 152.]

Joseph, M., Kearns, M., Morgenstern, J., and Roth, A. (2016). Fairness in learning: Classic and contextual bandits. In *NIPS*. [Pages 128 and 152.]

Kallus, N. and Zhou, A. (2019). The fairness of risk scores beyond classification: Bipartite ranking and the XAUC metric. *NeurIPS*. [Pages 27 and 28.]

Kay, M., Matuszek, C., and Munson, S. (2015). Unequal representation and gender stereotypes in image search results for occupations. In *CHI*. ACM. [Pages 7 and 155.]

Khattab, O. and Zaharia, M. (2020). Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*. [Page 14.]

Kilbertus, N., Carulla, M. R., Parascandolo, G., Hardt, M., Janzing, D., and Schölkopf, B. (2017). Avoiding discrimination through causal reasoning. In *NIPS*, pages 656–666. [Page 25.]

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR*. [Page 113.]

Kleinberg, J., Mullainathan, S., and Raghavan, M. (2017). Inherent trade-offs in the fair determination of risk scores. In *ITCS*. [Page 25.]

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. In *NIPS*, pages 4069–4079. [Page 25.]

Lahoti, P., Gummadi, K. P., and Weikum, G. (2019). Operationalizing individual fairness with pairwise fair representations. *VLDB Endowment*. [Page 27.]

Lang, J. and Rothe, J. (2016). Fair division of indivisible goods. In *Economics and Computation*, pages 493–550. Springer. [Page 29.]

Levy, R. (2021). Social media, news consumption, and polarization: Evidence from a field experiment. *American economic review*, 111(3):831–70. [Page 155.]

Li, S., Lattimore, T., and Szepesvári, C. (2019). Online learning to rank with features. *ICML*. [Page 91.]

Liu, L., Dean, S., Rolf, E., Simchowitz, M., and Hardt, M. (2018). Delayed impact of fair machine learning. In *ICML*. [Page 90.]

Liu, T.-Y. et al. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331. [Page 19.]

Marcus, M. and Ree, R. (1959). Diagonals of doubly stochastic matrices. *The Quarterly Journal of Mathematics*, 10(1):296–302. [Page 41.]

Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., and Diaz, F. (2018). Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *CIKM*. ACM. [Pages 5, 26, 34, 54, and 150.]

Mitra, B., Craswell, N., et al. (2018). *An introduction to neural information retrieval*. Now Foundations and Trends. [Page 14.]

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *ICML*, pages 1928–1937. [Page 70.]

Morik, M., Singh, A., Hong, J., and Joachims, T. (2020). Controlling fairness and bias in dynamic learning-to-rank. In *SIGIR*. [Pages 12, 84, and 152.]

Moulin, H. (2003). *Fair division and collective welfare*. MIT press. [Page 29.]

Nabi, R. and Shpitser, I. (2018). Fair inference on outcomes. *AAAI*. [Page 25.]

Nallapati, R. (2004). Discriminative models for information retrieval. In *SIGIR*. [Page 19.]

Nandy, P., Diciccio, C., Venugopalan, D., Logan, H., Basu, K., and Karoui, N. E. (2021). Achieving fairness via post-processing in web-scale recommender systems. [Pages 34 and 149.]

Narasimhan, H., Cotter, A., Gupta, M., and Wang, S. (2020). Pairwise fairness for ranking and regression. In *AAAI*. [Pages 27, 28, and 130.]

Nogueira, R. and Cho, K. (2019). Passage re-ranking with bert. *ArXiv*, abs/1901.04085. [Page 14.]

Nunley, J. M., Owens, M. F., and Howard, R. S. (2011). The effects of information and competition on racial discrimination: Evidence from a field experiment. *Journal of Economic Behavior & Organization*, 80(3):670–679. [Page 32.]

Nunziato, D. C. (2009). *Virtual freedom: net neutrality and free speech in the Internet age*. Stanford University Press. [Page 33.]

Otterbacher, J., Bates, J., and Clough, P. (2017). Competent men and warm women: Gender stereotypes and backlash in image search results. In *SIGCHI*, pages 6620–6631. [Page 155.]

Pan, B., Hembrooke, H., Joachims, T., Lorigo, L., Gay, G., and Granka, L. (2007). In google we trust: Users' decisions on rank, position, and relevance. *Journal of computer-mediated communication*, 12(3):801–823. [Page 33.]

Pariser, E. (2011). *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin. [Page 155.]

Patil, V., Ghalme, G., Nair, V., and Narahari, Y. (2020). Achieving fairness in the stochastic multi-armed bandit problem. *AAAI*. [Page 152.]

Plackett, R. L. (1975). The analysis of permutations. *Applied Statistics*, pages 193–202. [Page 68.]

Radlinski, F., Bennett, P. N., Carterette, B., and Joachims, T. (2009). Redundancy, diversity and interdependent document relevance. In *ACM SIGIR Forum*, volume 43, pages 46–52. [Pages 30 and 31.]

Radlinski, F. and Joachims, T. (2006). Minimally invasive randomization for collecting unbiased preferences from clickthrough logs. In *AAAI*, pages 1406–1412. [Page 91.]

Radlinski, F., Kleinberg, R., and Joachims, T. (2008). Learning diverse rankings with multi-armed bandits. In *ICML*. [Pages 30 and 91.]

Raff, A. (2009). Search, but you may not find. *New York Times*. [Pages 8 and 34.]

Richardson, M., Dominowska, E., and Ragno, R. (2007). Predicting clicks: Estimating the click-through rate for new ads. In *WWW*, pages 521–530. [Page 50.]

Robertson, J. and Webb, W. (1998). *Cake-cutting algorithms: Be fair if you can*. CRC Press. [Page 30.]

Robertson, S. E. (1977). The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304. [Pages 1, 4, 14, 38, 60, 95, 131, and 150.]

Rosen, S. (1981). The economics of superstars. *The American economic review*, 71(5):845–858. [Page 5.]

Rosenblat, A., Levy, K. E., Barocas, S., and Hwang, T. (2017). Discriminating tastes: Uber's customer ratings as vehicles for workplace discrimination. *Policy & Internet*, 9. [Page 32.]

Salganik, M. J., Dodds, P. S., and Watts, D. J. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. *Science*. [Pages 85 and 87.]

Salton, G. and McGill, M. J. (1983). *Introduction to modern information retrieval*. McGraw-Hill. [Page 14.]

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18. [Page 14.]

Saracevic, T. (1996). Relevance reconsidered. In *Proceedings of the second conference on conceptions of library and information science (CoLIS 2)*, volume 201, page 18. ACM New York. [Page 17.]

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. (2016).

Recommendations as treatments: Debiasing learning and evaluation. In *ICML*, pages 1670–1679. [Pages 48 and 90.]

Scott, M. (2017). Google fined record $2.7 billion in e.u. antitrust ruling. *New York Times*. [Page 2.]

Sharma, A., Hofman, J. M., and Watts, D. J. (2015). Estimating the causal impact of recommendation systems from observational data. In *EC*, New York, NY, USA. Association for Computing Machinery. [Page 152.]

Singh, A. and Joachims, T. (2017). Equality of opportunity in rankings. In *NIPS Workshop on Prioritizing Online Content (WPOC)*. [Page 26.]

Singh, A. and Joachims, T. (2018). Fairness of exposure in rankings. In *KDD*. [Pages 12, 26, 35, 48, 59, 76, 81, 86, 88, 90, 97, 110, 119, and 130.]

Singh, A. and Joachims, T. (2019). Policy learning for fairness in ranking. In *NeurIPS*. [Pages 12, 58, 86, 90, and 130.]

Singh, A., Kempe, D., and Joachims, T. (2021). Fairness in ranking under uncertainty. *arXiv preprint arXiv:2107.06720*. [Pages 12 and 121.]

Su, F. E. (2000). Cake-cutting algorithms: Be fair if you can. *The American Mathematical Monthly*, 107(2):185–188. [Page 29.]

Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*. MIT press. [Page 68.]

Swaminathan, A. and Joachims, T. (2015). Batch learning from logged bandit feedback through counterfactual risk minimization. *Journal of Machine Learning Research (JMLR)*. [Page 151.]

Sweeney, L. (2013). Discrimination in online ad delivery. *Queue*, 11(3):10:10–10:29. [Page 32.]

Tabibian, B., Gomez, V., De, A., Schölkopf, B., and Rodriguez, M. G. (2020). On the design of consequential ranking algorithms. In *UAI*. [Page 90.]

Taylor, M., Guiver, J., Robertson, S., and Minka, T. (2008). Softrank: optimizing non-smooth rank metrics. In *WSDM*. ACM. [Pages 20, 69, 91, and 128.]

Tennenholtz, M. and Kurland, O. (2019). Rethinking search engines and recommendation systems: a game theoretic perspective. *Commun. ACM*, 62(12):66–75. [Page 153.]

Varian, H. R. (1987). Fairness. *The New Palgrave: a Dictionary of Economics*, *London: Macmillan*, pages 275–76. [Page 29.]

Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*. [Page 90.]

Wang, X., Golbandi, N., Bendersky, M., Metzler, D., and Najork, M. (2018). Position bias estimation for unbiased learning to rank in personal search. In *WSDM*. ACM. [Pages 94 and 98.]

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256. [Pages 68, 69, 70, and 71.]

Williams, R. J. and Peng, J. (1991). Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268. [Page 70.]

Woodworth, B., Gunasekar, S., Ohannessian, M. I., and Srebro, N. (2017). Learning non-discriminatory predictors. In *Conference on Learning Theory*, pages 1920–1953. PMLR. [Page 25.]

Xiao, L., Min, Z., Yongfeng, Z., Zhaoquan, G., Yiqun, L., and Shaoping, M. (2017). Fairness-aware group recommendation with pareto-efficiency. In *RecSys*. [Page 28.]

Yadav, H., Du, Z., and Joachims, T. (2021). Policy-gradient training of fair and unbiased ranking functions. [Page 86.]

Yang, K. and Stoyanovich, J. (2017). Measuring fairness in ranked outputs. *SSDBM*. [Pages 26, 54, and 130.]

Yao, S. and Huang, B. (2017). Beyond parity: Fairness objectives for collaborative filtering. In *NeurIPS*. [Page 28.]

Ye, J., Chow, J.-H., Chen, J., and Zheng, Z. (2009). Stochastic gradient boosted distributed decision trees. In *CIKM*. [Pages 78 and 79.]

Yin, H., Cui, B., Li, J., Yao, J., and Chen, C. (2012). Challenging the long tail recommendation. *VLDB*. [Page 90.]

Yue, Y. and Guestrin, C. (2011). Linear submodular bandits and their application to diversified retrieval. In *NeurIPS*, pages 2483–2491. [Page 150.]

Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. (2017). Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*, pages 1171–1180. [Page 25.]

Zehlike, M., Bonchi, F., Castillo, C., Hajian, S., Megahed, M., and Baeza-Yates, R. (2017). Fa* ir: A fair top-k ranking algorithm. *CIKM*. [Pages 26, 54, and 130.]

Zehlike, M. and Castillo, C. (2020). Reducing disparate exposure in ranking: A learning to rank approach. In *The Web Conference*. [Pages 26, 59, 77, 81, and 86.]

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. (2013). Learning fair representations. In *ICML*, pages 325–333. [Page 25.]

Zhai, C. X., Cohen, W. W., and Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *SIGIR*, pages 10–17. ACM. [Page 70.]

Zhang, y. (2005). *Bayesian Graphical Model for Adaptive Information Filtering*. Phd dissertation, Carnegie Mellon University. [Page 44.]

Zhu, X., Goldberg, A., Van Gael, J., and Andrzejewski, D. (2007). Improving diversity in ranking using absorbing random walks. In *HLT:NAACL*, pages 97–104. [Page 30.]

Zliobaite, I. (2015). On the relation between accuracy and fairness in binary classification. *FATML Workshop at ICML*. [Pages 25, 26, and 46.]

Zoghi, M., Tunys, T., Ghavamzadeh, M., Kveton, B., Szepesvari, C., and Wen, Z. (2017). Online learning to rank in stochastic click models. In *ICML*. [Page 91.]