Ranking for Decision Making: Fairness and Usability

by

Caitlin Kuhlman

A Dissertation

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Computer Science

by

May, 2020

APPROVED:

Professor Elke A. Rundensteiner Worcester Polytechnic Institute Advisor	Professor Lane Harrison Worcester Polytechnic Institute Committee Member
Professor Randy Paffenroth Worcester Polytechnic Institute Committee Member	Professor Carlos Scheidegger University of Arizona External Committee Member
Professor Craig Wills Worcester Polytechnic Institute Head of Department	

Abstract

Today, ranking is the de facto way that information is presented to users in automated systems, which are increasingly used for high stakes decision making. Such ranking algorithms are typically opaque, and users don't have control over the ranking process. When complex datasets are distilled into simple rankings, patterns in the data are exploited which may not reflect the user's true preferences, and can even include subtle encodings of historical inequalities. Therefore it is paramount that the user's preferences and fairness objectives are reflected in the rankings generated. This research addresses concerns around fairness and usability of ranking algorithms. The dissertation is organized in two parts.

Part one investigates the usability of interactive systems for automatic ranking. The aim is to better understand how to capture user knowledge through interaction design, and empower users to generate personalized rankings. A detailed requirements analysis for interactive ranking systems is conducted. Then alternative preference elicitation techniques are evaluated in a crowdsourced user study. The study reveals surprising ways in which collection interfaces may prompt users to organize more data, thereby requiring minimal effort to create sufficient training data for the underlying machine learning algorithm. Following from these insights, RanKit is presented. This system for personalized ranking automatically generates rankings based on user-specified preferences among a subset of items. Explanatory features give feedback on the impact of user preferences on the ranking model and confidence of predictions. A case study demonstrates the utility of this interactive tool.

In part two, metrics for evaluating the fairness of rankings are studied in depth, and a new problem of fair ranking by consensus is introduced. Three group fairness metrics are presented: *Rank Equality, Rank Calibration,* and *Rank Parity* which cover a broad spectrum of fairness considerations from proportional representation to error rate similarity across groups. These metrics

are designed using a pairwise evaluation strategy to adapt algorithmic fairness concepts previously only applicable for classification. The metrics are employed in the FARE framework, a novel diagnostic tool for auditing rankings which exposes tradeoffs between different notions of fairness. Next, different ways of measuring a single definition of fairness are evaluated in a comparative study of state-of-the-art statistical parity metrics for ranking. This study identifies a core set of parity metrics which all behave similarly with respect to group advantage, reflecting well an intuitive definition of unfairness. However, this analysis also reveals that under relaxed assumptions about group advantage, different ways of measuring group advantage yield different fairness results. Finally, I introduce a new problem of fair ranking by consensus among multiple decision makers. A family of algorithms are presented which solve this open problem of guaranteeing fairness for protected groups of candidates, while still producing a good aggregation of the base rankings. Exact solutions are presented as well as a method which guarantees fairness with minimal approximation error. Together, this research expands the utility of ranking algorithms to support fair decision making.

Acknowledgements

I am deeply grateful to my advisor, Professor Elke Rundensteiner, for guiding and encouraging me through my PhD studies. I appreciate your unfailing generosity with your time and attention, chasing down many late night deadlines with me despite a superhuman workload, and helping me through many moments of uncertainty. I have been fortunate to have your mentorship and support while following new research directions and forging an interesting and meaningful topic for this dissertation.

I thank Professor Lane Harrison for many great conversations and guidance on research in HCI and visual analytics. Special thanks also go the members of my committee Professor Randy Paffenroth and Professor Carlos Scheidegger for valuable feedback and discussion at many stages of developing this dissertation. I appreciate your encouragement and input which pushed this work forward in multiple ways. I also thank Professor George Heineman for his feedback during my qualifier research.

I thank and acknowledge my collaborators at WPI: Lei Cao, Yizhou Yang, Rodica Neamtu, Ramoza Ahsan, Erin Teeple, MaryAnn VanValkenburg, Walter Gerych, and the many undergraduate students I had the pleasure of working with on the MATTERS and RanKit systems. A big shoutout to all the members of the DSRG community, and to my WiDS co-ambassadors Cansu Sen and Melanie Jutras. Thanks also to Latifa Jackson the BPDM crew for many great conversations and experiences. Thank you to Molly Duggan who was my first peer role model in tech and Joseph Paul Cohen at UMass Boston whose infectious enthusiasm for research got me started on this journey in the first place.

Thank you to Chris Anderson at the Massachusetts High Tech Council for funding me to work on the MATTERS project over several years, and to Oak Ridge National Lab and NSRDEC, the Department of Education, and WPI for additional funding. Thanks also to Leonid Liebman at MITRE Corp, and to Kush Varshney and the Science for Social Good team at IBM Research for excellent internship experiences.

Finally, I couldn't have done this without loving family and friends who have ridden the highs and lows of these years with me. Thank you to Emily for being there the many times I needed my sister and friend. Especially thank you to my parents, Louise and Mike, tireless champions of their daughter the perpetual student. You encouraged and nurtured my curiosity all my life while demonstrating open mindedness, work ethic, and creative problem solving. Your emotional and practical support has afforded me a luxury of time to study both art and science, and I am grateful.

What a gift it has been to come this far with you all and I look forward to the things we will accomplish together in the future.

Publications

Topic I: Interactive Ranking Analytics

1. Caitlin Kuhlman, Paul-Henry Schoenhagen, MaryAnn VanValkenburg, Diana Doherty, Malika Nurbekova, Goutham Deva, Zarni Phyo, Elke Rundensteiner, Lane Harrison. *Evaluating Preference Collection Methods for Interactive Ranking Analytics*, CHI 2019.

   Relationship to this dissertation: In this work we conduct a crowdsourced user study comparing preference elicitation methods for personalized ranking. The study is presented in Chapter 3.

2. Caitlin Kuhlman, MaryAnn VanValkenburg, Diana Doherty, Malika Nurbekova, Goutham Deva, Zarni Phyo, Elke Rundensteiner, Lane Harrison. *Preference-driven Interactive Ranking System for Personalized Decision Support*, CIKM 2018. (demo)

   Relationship to this dissertation: This work demonstrates RANKIT, a mixed initiative interactive system for ranking, along with several case studies, discussed in Chapter 4.

Topic II: Ranking for Fair Decision Making

3. Caitlin Kuhlman, MaryAnn VanValkenburg, Elke Rundensteiner, *FARE: Diagnostics for Fair Ranking using Pairwise Error Metrics*, WWW 2019.

   Relationship to this dissertation: In this work, we introduce three pairwise fairness metrics for ranking, presented in Chapter 7, and the FARE framework for auditing rankings in Chapter 7.2.

4. Caitlin Kuhlman, Walter Gerych, Elke A. Rundensteiner, *Measuring Group Advantage: A Comparative Study of Fair Ranking Metrics*, in submission to a major

conference 2020.

Relationship to this dissertation: This work surveys statistical parity metrics for ranking, providing a conceptual lens for metric comparison and revealing relationships and tradeoffs among different ways of measuring group advantage in rankings. This work is discussed in Chapter 8.

5. Caitlin Kuhlman, Elke A. Rundensteiner, *Rank Aggregation Algorithms for Fair Consensus*, in submission to a major conference 2020.

   Relationship to this dissertation: We introduce a new problem of incorporating fairness criteria for the candidates being ranked when aggregating multiple rankings, along with exact and approximate solutions. This work is presented in Chapter 9.

Other Publications

Other research I have undertaken during my PhD at WPI include the following publications on the topic of outlier detection in big data.

6. Caitlin Kuhlman, Yizhou Yan, Lei Cao, Elke Rundensteiner, *Pivot-based Distributed K-Nearest Neighbor Mining*, ECML-PKDD 2017.

7. Yizhou Yan, Lei Cao, Caitlin Kuhlman, Elke Rundensteiner, *Distributed Local Outlier Detection in Big Data*, KDD 2017.

8. Lei Cao, Yizhou Yan, Caitlin Kuhlman, Qingyang Wang, Elke Rundensteiner, Mohamed Eltabakh, *Multitactic Distance-based Outlier Detection*, ICDE 2017.

During my PhD I have published on topics related to data-driven decision making with collaborators within and outside WPI.

9. Caitlin Kuhlman, Latifa F. Jackson, Rumi Chunara, *No computation without representation: Avoiding data and algorithm biases through diversity*, EDSC 2020.

10. Latifa F. Jackson, Caitlin Kuhlman, Fatimah L.C. Jackson, Keolu Fox, *Including Vulnerable Populations in the Assessment of Data from Vulnerable Populations*, Frontiers in Big Data 2019.

11. Rodica Neamtu, Caitlin Kuhlman, Ramoza Ahsan, Elke Rundensteiner, *The impact of Big Data on making evidence-based decisions*, Frontiers in Data Science, 2017.

12. Caitlin Kuhlman, Karthikenyan Natesan Ramamurthy, Prassana Sattigeri, Aurelie C. Lozano, Lei Cao, Chandra Reddy, Aleksandra Mojsilovic, Kush R. Varshney *How to foster innovation: a data-driven approach to measuring economic competitiveness*, IBM Journal of Research and Development 2017.

Contents

List of Figures

List of Tables

1

Introduction

Ranking is commonly used to prioritize among candidates for desirable outcomes like jobs, loans, or educational opportunities. For such high-impact applications, fairness concerns are often paramount, whether enforced by legal standards (e.g., the 80% rule in discrimination law [61]) or by internal policies of an organization aiming to ensure diversity. Unfortunately, people performing ranking analysis may suffer from implicit bias in their decision making [74], which has had a demonstrated negative impact for critical tasks such as hiring [16, 20, 145].

Complicating matters further, increasingly the judgments of human analysts are augmented by decision support tools or even fully automated screening procedures which rank candidates [15, 37, 70, 150]. Such systems may encode unfair bias, perhaps present in the training data [135], reflected by the design of scoring functions [10], or due to differences in the way of members of different groups represent themselves [6]. Applied for search, recommendation, and indexing, this process is opaque and it is difficult to assess its impact on our decision making. This new interplay of human bias and machine learning may further impede equitable decision making in unforeseen ways.

As an example of the potential for automating discrimination in such systems, consider that Amazon recently revealed a failed attempt to design a hiring algorithm to screen and rank candidates. The project was dropped when they found that the model inadvertently encoded a gender bias against women [49]. For a regulated domain like employment, but also education, housing, and many others, such practices are illegal in the United States [12]. In this case the unfair bias was caught, yet the development of ranking algorithms for hiring is widespread [3], while no systematic approaches to audit these methods are available to date.

To address this important societal problem, recent research focuses on the design of

metrics for measuring unfair bias in individual rankings and strategies for mitigating its effect [10, 18, 34, 70, 106, 141, 155, 161]. For instance, LinkedIn recently incorporated a fairness framework into their Talent Search feature that helps recruiters find job candidates [70]. The continued study of fairness metrics for rankings is crucial for the development of fair socio-technical systems. Meta-analysis is also required, to provide guidance on the choice of appropriate evaluation metric for different contexts, and to expose tradeoffs between different notions of what is fair.

Nuanced analysis and interpretation is required when considering the implications of automated decision making in our society. For instance, fairness criteria are often cast as competing with things like safety, performance, or other "meritocratic" measures [44]. However, it may be the case that the values we are optimizing for are the very things that perpetuate and impose structural inequality [122]. Oversimplification of complex problems can hide potentially dangerous assumptions, and may encode inherent bias in the underlying data used to train a ranking model. As another example, consider college rankings published online and used by students and faculty. Highly ranked colleges often have poor outcomes for low-income students, such as lower graduation rates or excessive debt after graduation [146]. The rankings could be considered unfair for only providing utility to high-income students. Left unchecked, such harm compounds through the creation of negative feedback loops. Colleges consistently given a low rank will attract less talent, decreasing their potential to improve [71, 124]. These institutions could become entrenched in the position determined by the ranking model - often proprietary and not disclosed.

The vast potential for harm in such cases highlights the need for open and transparent procedures to audit and correct for unfair bias in rankings. This requires the design of error metrics appropriate for detecting unfair group outcomes in rankings. Furthermore, it is imperative that we design highly useable systems that empower people to access technologies to answer their own questions and understand the impact of their own priorities on decision outcomes. To help avoid unfair practices and provide and utility to decision makers, this dissertation addresses concerns around fairness and usability of ranking algorithms.

1.1 Dissertation Organization

The dissertation is organized in two parts. Part I investigates the usability of ranking as a tool for complex, real-world decision making through the development and evaluation

of interactive visual analytics systems for ranking. Chapter 2 reviews the state-of-the-art for multiple relevant aspects of interactive ranking, and provides a detailed requirements analysis for mixed-initiative ranking systems. Chapter 3 presents a crowdsourced user study evaluating preference collection methods for interactive ranking. Informed by these investigations, Chapter 4 then gives an overview of RanKit, a general system for interactive ranking analytics. Chapter 5 reviews related work on interactive systems for ranking.

Part II then presents a detailed study of group fairness in rankings. Chapter 6 first provides background on the topic of algorithmic fairness and state-of-the-art fairness metrics for ranking and formalizes the fair ranking problem. Chapter 7 then introduces our proposed pairwise metrics for evaluating various fairness criteria for rankings, and the FARE framework for auditing the fairness of rankings. FARE exposes tradeoffs between metrics which evaluate different notions of fairness. In Chapter 8 we then further investigate metrics which evaluate a single fairness definition – Statistical Parity – in a comparative study. In Chapter 9 we define a new problem of fair rank aggregation. A family of algorithms for exact and approximate solutions is presented. Chapter 10 discusses related work for fair ranking.

Chapter 11 concludes by sketching possible avenues for future work in visual analytics for fair ranking which build on the areas of study in this dissertation.

Part I

Interactive Ranking Analytics

2

Introduction: Mixed-initiative Ranking Systems

For decision making when the number of factors impacting choice is large, people often consult rankings. A ranking can distill high dimensional information into a simple ordered list, helping people to quickly grasp the relative merit of objects or choices. People rely on rankings published by companies, consumer groups, and government agencies for guidance across many domains - from consumer choices of products and services [71], to pivotal life decisions such as college choice [87, 164], to evaluating the economic competitiveness of different regions [51, 139]. Such rankings are typically consumed "off the shelf", lacking any personalization to reflect the priorities of individual users, and using ranking procedures which are opaque to the user.

At the same time, powerful learning-to-rank algorithms [111] are used extensively in web-based search and recommendation to provide personalized rankings to users (in terms of top search results). Indeed, our everyday experiences are increasingly shaped by powerful algorithms which produce such rankings over large data sets. Recently, interactive systems have been proposed [72, 105, 147] which aim to assist users in creating personalized rankings for their own decision making. Rather than consuming rankings in the form of prescriptive evaluations, these systems allow users to control the ranking process by specifying their personal preferences. This information is then used by the system to produce a global ranking ranking over the entire dataset.

Interactive ranking systems rely on user-machine collaboration to facilitate sense-making that would otherwise not be possible. Machine automation provides computational power to accomplish tasks laborious or impossible for a user, while humans provide domain expertise, understanding of the task at hand, and personalized preferences. Inter-

actions include manual adjustments of attribute weights, as well as specifying preferences over items being ranked. Mixed-initiative systems [86] employ machine learning to learn a global ranking from these interactions. A typical use case for an interactive ranking system is a personalized college ranker. A student can use such a system to specify their preferences over colleges they have visited so far, based on their own goals and interests. The system then automatically generates a global ranking over a larger set of universities that the student is not able to visit in person. This process may provide further insight by revealing the data attributes used by the system to create the ranking – thus allowing the student to better understand their own priorities. We revisit this example in our case study evaluation in Section 4.3.

2.1 Challenges in System Design

The interplay of user interactions and automation must be carefully considered in the design of mixed-initiative systems to allow the user to guide the automation effectively and derive a truly valuable result. A key challenge in for interactive ranking lies in the elicitation of preference information from users. While preference elicitation techniques have been considered in the context of interactive recommender systems [2, 82, 99, 100], their use in interactive ranking systems has not been formally evaluated. The impact of different preference collection mechanisms on user behavior and level of satisfaction with the ranking system is thus not well understood. Further, the specification of enough information so that the learning engine can reliably infer a useful ranking represents an arduous task for humans. Yet the availability of a large enough training dataset over which to learn a ranking is pivotal in determining a meaningful ranking [151]. As discussed by Crouser et al. in [46], for the design of appropriate systems we should evaluate and quantify both the computational complexity of the processes used as well as the complexity of the human effort itself. A final concern is that uncertainty inherent in visual analytics systems has the potential erode users' trust in the model, as examined by Sacha et al. [134]. Communicating to the user to ensure their understanding and confidence in the ranking process. Initial proposed systems do not provide guidance to the user on the quality of the learned ranking for decision making.

To address these challenges, in part one of this dissertation we aim to better understand the design and usability of mixed-initiative systems for ranking which are designed to allow a single user to fully control the ranking process by explicitly specifying their personal preferences. An initial requirements analysis for rank visualization was provided

by Gratzl et al. in [72] to motivate the design of the Lineup system, a manual tool for rank building. Next we build on this analysis with an in-depth requirements analysis for interactive ranking systems which incorporate machine learning to automate the ranking process. Additional considerations reflect principles for mixed-initiative system design laid out by Horowitz [86], and guidelines for handling uncertainty in visual analytics systems.

2.2 Requirements Analysis for Interactive Ranking

2.2.1 R1: Provide Significant Value-Added Benefit Through Automation.

Following directly from the requirements laid out in [86], the ranking engine for mixed initiative systems must be able to learn models that contribute meaningful insights to the ranking process. This machine learning problem is different from conventional supervised learning-to-rank settings in that there is no absolute "ground truth" ranking with which to train a model. When we ask the user to input their knowledge about the dataset, we cannot expect them to label objects with an exact position in the final ranking, as might be provided in a traditional training dataset. If users were capable of performing this task, they would not need to use an analytics system to aid their ranking process. Instead, we expect that users will impart partial knowledge from which a ranking is learned and then applied *over the same dataset* in a semi-supervised manner [144]. This interaction helps the user gain a global understanding of an entire dataset. In the process they may also come to understand their own intuition and preferences better, similar to how systems for personal informatics encourage self-reflection [109].

To evaluate the value added through automation, we must consider metrics appropriate for 1) evaluating the performance of the ranking model, and 2) determining the ability of the system to provide a ranking which satisfies the user. To address the former, we can adopt approaches from active and semi-supervised learning for model evaluation over a partially labeled dataset. An important consideration is estimating the "sample complexity" of the ranking problem, i.e. the number of labeled training items needed to outperform a random ordering in expectation [64, 132, 151]. Evaluating the latter hinges on subjective evaluation by the user. Rankings are often used to capture an ill-defined or complex quality, which is difficult to measure directly. For instance, in the context of college rankings no one can say objectively that Harvard is a better choice than Yale,

since the best choice will vary across individual students. HCI evaluation strategies are required to evaluate how well the learned rankings reflect the user's input.

2.2.2 R2: Capture Meaningful, Unbiased User Knowledge

The driving assumption behind the automated ranking process is that the user has partial knowledge about objects in the dataset, from which a satisfactory global model can be inferred. R1 therefore depends strongly on the ability of the user to specify their preferences among the items in the dataset effectively. Ranking systems should capture the users' partial understanding of the problem in as intuitive a manner as possible. Further, interaction mechanisms should be carefully designed taking the human analyst's cognitive ability into consideration. For instance, it has been shown that humans may be more cognitively adept at making relative judgments [30, 38] rather than assigning explicit value judgments. Interaction modalities should also capture accurate information, unbiased toward any particular preference. For instance, display order can impose a position bias on the choices made by users [75, 91, 159], as has been observed in web search results.

2.2.3 R3: Avoid Excessive User Specification Effort

As discussed by Crouser et al. in [46], for the design of appropriate mixed-initiative systems we should quantify both the computational complexity of the processes used, as well as the complexity of the human effort itself. For interactive ranking, the user effort involved in providing input to the system directly impacts the potential quality of the ranking model. Machine learning ranking models require many labeled examples, but too much tedious effort may prohibit user engagement [99]. Well-designed interfaces are necessary to extract as much information as possible from minimal user input.

Existing interactive ranking systems [104, 105, 147] leverage *pairwise* learning-to-rank algorithms, which reduce the problem of ranking a set of objects to the simpler binary classification task over ordered pairs of objects [111]. According to sample complexity results for active ranking, the number of labeled pairs required to establish a meaningful ordering over a given set of items is quadratic in the number of items being ranked [151]. For even moderately sized datasets, the utility of such an approach could quickly be outweighed by the burden on the user having to specify so many pair wise comparisons. Therefore, user effort is a major concern in the design of visual analytics systems which rely on pairwise learning-to-rank algorithms.

2.2.4 R4: Foster Trust by Exposing Uncertainty

Finally, for human-machine collaboration to be effective in real-world systems, the user must trust the predictions generated by the system. Tracking uncertainty as it propagates through a visual analytics system has been identified as crucial for this task by Sacha et al. [134]. This uncertainty should be communicated to the user to ensure their confidence in the ranking process. For automatic ranking, uncertainty may be present at multiple endpoints in the system, including:

1. *Uncertainty in user preferences.* The intuition or domain knowledge users have regarding the importance of data attributes may conflict with their opinions about the relative merit of individual items. For example, in the context of college rankings, a person may believe that college A is better than college B due to its reputation, even though they value and must strive for affordability and college B is much more expensive than college A. In this case, a trade-off between favoring one type of user preference over the other arises in the global ranking model. It has also been observed that when collecting preferences for recommendation, user ratings of items are likely to be inconsistent or inaccurate [7]. Therefore in designing interactive systems we cannot always count on the user to be a perfect oracle.

2. *Disagreement between user input and the model.* A user may think that their preference information poses hard constraints on the model, and that whatever order they set for the training items will always be reflected in the global ordering. This is not the case – in fact the learning task is to match the partial input as close as possible, with minimal changes. In general, it is not guaranteed that a weighted attribute formula can be learned that matches the user's preferences completely. This may be confusing or frustrating (as observed by Wall et al. in the Podium system [147]), and negatively impact user engagement and trust.

3. *Uncertainty in the model.* As discussed in **R1** and **R3**, the quality of the generated ranking model depends heavily on the amount of training data provided. Given only a few pairs, the global ranking may not perform much better than a random permutation. If the user is not made aware that the system does not have sufficient information to render a clear ordering, they may not believe that the ranking process is working effectively. Even with sufficient training examples, it is possible that multiple rankings are equally likely as a result. In this case, the set of possible outcomes should be available to the user.

3

Evaluating Preference Collection Methods for Interactive Ranking

As a first step toward meeting the requirements for mixed-initiative ranking systems, this chapter presents an in-depth evaluation of different preference collection modes for interactive ranking. We compare the impact of alternative interfaces which allow users to *explicitly* specify their preferences over items in the dataset using *relative judgments* (requirement **R2**). We evaluate both the performance of the machine learning algorithm (**R1**) as well as user effort and behavior (**R3**). The study was published in the research paper [102]:

> Caitlin Kuhlman, Diana Doherty, Malika Nurbekova, Goutham Deva, Zarni Phyo, Paul-Henry Schoenhagen, MaryAnn VanValkenburg, Elke Rundensteiner, and Lane Harrison. Evaluating Preference Collection Methods for Interactive Ranking Analytics. *In the CHI Conference on Human Factors in Computing Systems Proceedings* ACM, 2019

Three alternative methods for collecting preferences over items in the dataset are evaluated: *Sub-list Ranking*, *Categorical Binning*, and *Pairwise* preference collection methods. These interaction modes cover the spectrum of core methods that have been employed in interactive ranking and recommendation systems to date (Detailed in the Related Work in Chapter 5). Our study was conducted on the Mechanical Turk platform using a between subjects design in which each participant was randomly assigned to one of three preference collection modes. We implement each interaction mode as part of a mixed-initiative ranking system (described in 3.1.2) to compare people's interactions using each of these three conditions.

The contributions of the study include:

1. We design three alternative interfaces that embody the three distinct modes of preference specification embedded into an interactive ranking tool to provide the study subjects with an end-to-end experience.

2. We conduct a large scale ($n = 144$ subject) crowdsourced user study to evaluate the complexity of the human effort in interaction, sample complexity of the information extracted from interactions, and the impact on user satisfaction with the resulting ranking.

3. Our study finds that the categorical approach provides the best value-added benefit to users, requiring minimal effort, and encouraging them to provide statistically significant more training data, which positively impacts the underlying machine learning algorithm's ability to create a preferred ranking result.

4. Our findings on different effects of the alternate preference modes raise interesting questions requiring future investigation into the composition of interaction modes and alternative means for driving up user engagement in ranking systems.

3.1 Method of the Study

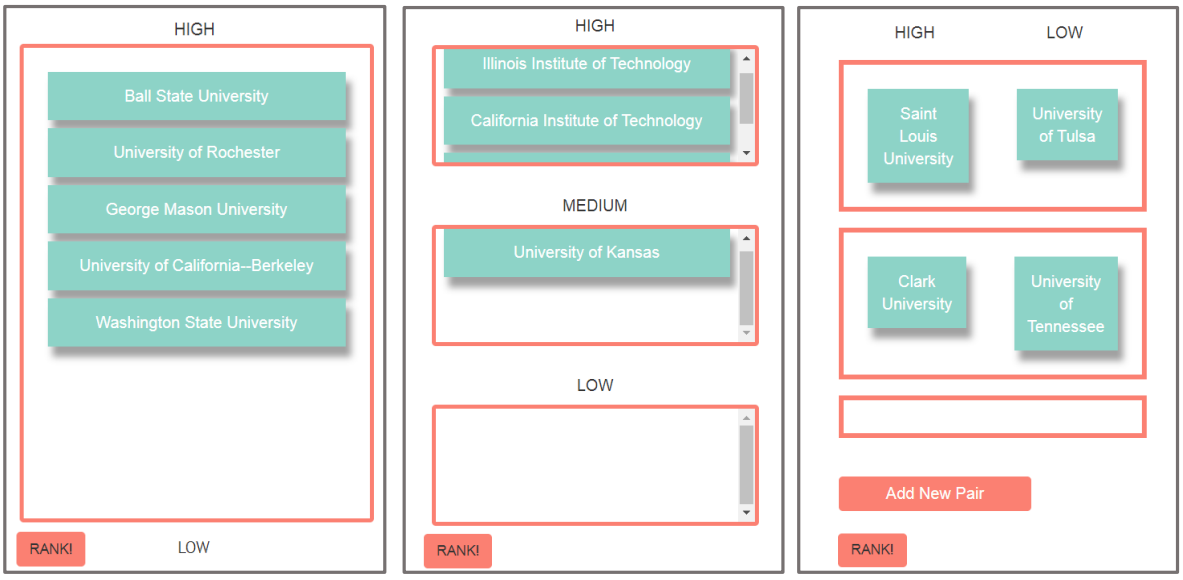
Research question investigated in the study are (1) do users behave differently depending on the interaction mode, (2) does the mode of interaction impact user satisfaction, and (3) what kind of trade off does each mode offer to balance user effort with the training requirements of the underlying ranking engine. Drawing on analytic approaches from several recent studies examining user behavior [22, 55, 66, 81, 148], and analysis of the computational complexity of system processes [46], we frame our research questions as follows:

- **Interaction behavior:** does the collection mode impact measures of behavior such as total time spent entering preferences, or the number of data items added?

- **Self-reported user experience:** does the collection mode affect the perceived ease of use of the ranking tool? Does it impact their anticipated adoption of the tool for ranking tasks?

- **System performance:** does the collection mode affect the size of the training data generated from user preferences?

3.1.1 Selection and Design of Alternate Preference Collection Methods



(a) Sub-list Ranking mode.

(b) Categorical Binning mode.

(c) Pairwise mode.

Figure 3.1: Alternative preference elicitation interfaces.

The three methods for collecting user preferences over items in the dataset chosen for the study are: *Sub-list Ranking*, *Categorical Binning*, and *Pairwise* preference collection methods. The use of pairwise comparisons has been popular for preference elicitation [30, 38, 113]. List comparison is used in the recently proposed visual analytics system, Podium [147], however we design our list to directly capture user preferences over a subset of items, and we do not infer item relationships implicitly. Finally, to allow users to group similar items as in previous ranking and recommendation systems [82, 105], we implement an interface where users group items into categories: high, medium, or low. The resulting collection interfaces are shown in Figure 3.1.

A *pairwise* learning-to-rank algorithm powers the ranking engine [91]. For this, pairs of items are extracted from user interactions and used to train the ranking model. Details of the machine learning process are given in Section 3.1.2. Each interaction mode is next described in detail.

Sub-list Ranking Preference Collection. The Sub-list preference collection mode (shown in Figure 3.1a) closely matches a typical ranking activity. Preferences are specified

by sorting a subset of objects into a completely ordered list. Items at the top of the list are preferred to those placed below. At a minimum, two objects must be placed in the list so as to form one pair. The user then can add any number of additional items up to specifying a complete ordering over all items.

Categorical Binning Preference Collection. The second preference collection method uses a categorical approach (3.1b). Here users express their preference by binning a subset of items into three categories: high, medium, low. Items within each category are not compared. However, items in the high category are preferred to all items in both the medium and low categories, and items in the medium category are preferred to all items in the low category. Users must specify at a minimum two items in separate categories in order to derive a ranking. The user may choose to organize objects in any two out of the three categories, or use all categories, with any number of objects in each.

Pairwise Preference Collection. The last ranking method we consider is the Pairwise preference collection mode (3.1c). Here, users directly express their preferences as binary relations between pairs of items. Users place items in ordered pairs, with the object on the left compared to the object on the right. Unlike the other comparison modes, in the pairwise interface the same object can be entered multiple times if it is preferred to multiple other items.

3.1.2 College Ranker Interactive Ranking Scenario

These alternative preference collection modes are incorporated into an interactive College Ranking system. The US News and World Report Best Colleges dataset[1] is used. The dataset contains both numeric and categorical attributes of colleges in the United States. The system is composed of two views, a "Build page" where users enter their preferences, and an "Explore page" where they view the global ranking generated based on their input. They can iterate between these views to continually refine their ranking. These views are shown in Figure 3.2 and described in detail as incorporated into a general mixed-initiative system for ranking in the next Chapter, Section 4.2.

Ranking Engine. Under the hood, a mixed-initiative system for ranking leverages a learning-to-rank machine learning algorithm to generate a global ranking of the dataset

[1]https://www.usnews.com/best-colleges/rankings/national-universities

from the partial input collected from the user. We employ the RankSVM algorithm [91] which uses a Support Vector Machine (SVM) to distinguish between correctly ordered and incorrect pairs of data objects. The key idea is an assumption of a linear function $U(x) = \vec{w}^T x$ where $\vec{w}$ is a d-dimensional weight vector mapping each object in the dataset x_i to a value corresponding to its rank. Then the following holds true:

$$\vec{w}^T x_i > \vec{w}^T x_j \implies \vec{w}^T (x_i - x_j) > 0$$

Therefore, instead of learning the ranking from the individual points x_i in the training dataset, the function can be learned over the combined feature vector $(x_i - x_j)$ of each ordered pair of objects. Each training pair is assigned a binary class label $c \in \{-1, 1\}$, where a label of 1 indicates a correctly ordered pair, and -1 indicates an inverted pair. The weight vector $\vec{w}$ is a hyperplane decision boundary which distinguishes between these two classes while maximizing the space between them. Once the boundary has been learned from the training data, a global ranking over all unseen data can be extracted. For each object x_i, $\vec{w}x_i$ gives a score $\hat{y}_i$ which determines its rank position.

In a typical *supervised learning* problem formulation, the true ranking over all n training data points is known. For interactive ranking, the problem is *semi-supervised* [144], in that labels are given only for a subset of $m < n$ data points. A key consideration for the performance of the ranking model in this setting is sample complexity analysis on the number of training pairs required to effectively learn a model. Wauthier et al. [151] consider the sample complexity of the RankSVM algorithm. They observe that if pairs are selected at random and labeled, then the RankSVM algorithm performs optimally and requires $O(n)$ pairs to produce a better than random expected result.

This complexity analysis is crucial to understanding whether interactive ranking can be effective given a small number of examples from a user. Clearly, for dataset of $n = 100$ items, having to manually specify preferences over 100 pairs puts a non-trivial burden on the user. To reduce this, elicitation techniques should generate as much information as possible for the least amount of user effort. The mode of preference collection employed impacts the number of pairs that can be extracted from the user input. We examine this effect in depth in Section 3.2.2.

3.1.3 Pilots and Experiment Planning

Participants were recruited through Amazon's Mechanical Turk (AMT) to participate in a college ranking task. Two pilot studies were conducted to refine experimental instructions

and pacing, and to estimate the required sample size for the final study. We conducted effect size and statistical power analyses. Specifically, we estimated the variance in our quantitative measures based on results pilot studies. These estimates were combined with the observed means to approximate how many participants were needed to ensure our experiments would reliably detect meaningful differences between the conditions.

Following Hara et al [79], our workers were paid $1.25 based on the average completion time of 5-8 minutes in our pilot studies. Payments were structured as $1.00 base rate with a bonus of $0.25 offered for creating a satisfactory ranking, to incentivize engagement with the tool. Unbeknownst to the workers, they all were paid the bonus. Average hourly wage for the full study was $10.42, exceeding US federal minimum wage of $7.25. Each participant was randomly assigned to one of the three interaction modes and rewards were consistent throughout the three methods. All participants viewed an IRB-approved consent form.

3.1.4 Procedure and Tasks

Our procedure consisted of four phases: *Training, Rank Building, Rank Exploration*, and *Post-test Survey*. Each phase is next described detail, and views of each phase are shown in Figure 3.2.

Training: We provided participants with an instruction page that briefly described their task and the interaction mechanisms in the ranking tool. For example, in the sub-list collection mode, the instructions stated:

> *On the next page, you will use an interactive tool to create a personalized college ranking. First, you will use the Build Tool to enter your preferences about colleges in the dataset. Choose as many colleges as you wish and place each one into a ranked list. After you make your selections, the system will provide you with a ranking of the entire dataset of colleges based on your initial choices.*

Each specific interaction mode was described in detail, with animated gifs illustrating the preference collection process. For the Sub-list mode users were instructed:

> *Drag colleges from the dataset on the left to the list on the right. Place the most preferred colleges at the top of the list, and the rest in descending order of your preference.*

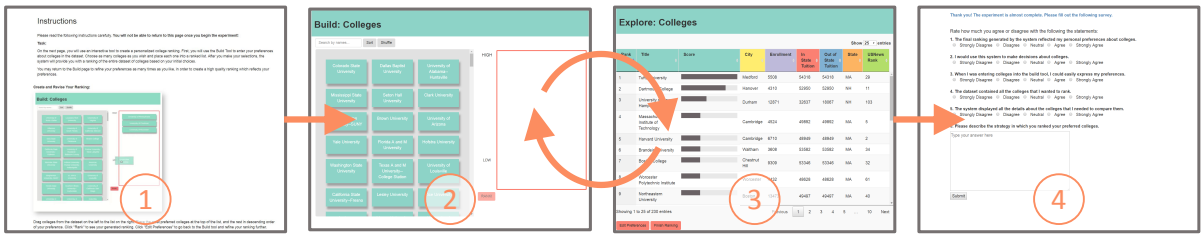


Figure 3.2: Experiment phases: (1) training, (2) rank building, (3) rank exploration, (4) post-test survey. Participants can iterate between build and explore unlimited times.

Rank Building: After viewing the instructions, users proceeded to the College Ranker interactive ranking tool. The Build page contained the randomly assigned preference collection interface. Participants were able to interact with the colleges in the dataset, entering as many preferences as they desired, without any time limit. When participants were satisfied with their preferences they could click a "Rank" button to advance to the Rank Exploration phase.

Rank Exploration: The Explore page displayed the generated ranking over the entire dataset in a tabular format. On this page users could explore the ranking by scrolling or paging through results, examining the order of items and the scores assigned by the ranking engine. From here users could click "Edit Preferences" button to navigate back to the previous build page and amend or refine their preferences. Users were able to iterate between the Build and Explore pages as many times as they wished. To complete the ranking task, users could click "Finish Ranking". A modal window prompted them "Would you like to revise your ranking by returning to the Build page?" to ensure users were aware of the option to return to build. Users could then click "Yes - Return to edit preferences" or "No - This is my final ranking", which advanced them to the final phase of the study.

Post-test Survey: Participants were provide with a short set of statements and asked to indicate their agreement to provide qualitative feedback.

3.1.5 Measures

To evaluate the three interaction modes, we collect a number of qualitative measures by logging user actions during the *Rank Building* and *Rank Exploration* phases of the study. We also record the time spent in each phase of the study. Interactions logged include:

- **Additions:** the number of items participants entered into the preference collection interface by dragging them from the data pool.
- **Removals:** the number of items participants removed from the preference collection interface and returned to the data pool.
- **Selections:** the set of items entered into the preference collection interface.
- **Ranks:** the number of times the user clicked the "Rank!" button to advance from the Build page to the Explore page.
- **Refines:** the number of times the user clicked the "Edit Preferences" button to return to the Build page from the Explore page.

To evaluate the system performance we consider the size of the training dataset provided to the ranking engine using each interface. As detailed in 3.1.2 the training data consists of pairs of data objects, generated from the preferences specified by the user. We measure the training data size in two ways:

- **Pair growth rate:** the number of pairs p generated from m items entered by the user. [1]
- **Actual pairs:** an empirical count of the number of training data pairs generated in practice.

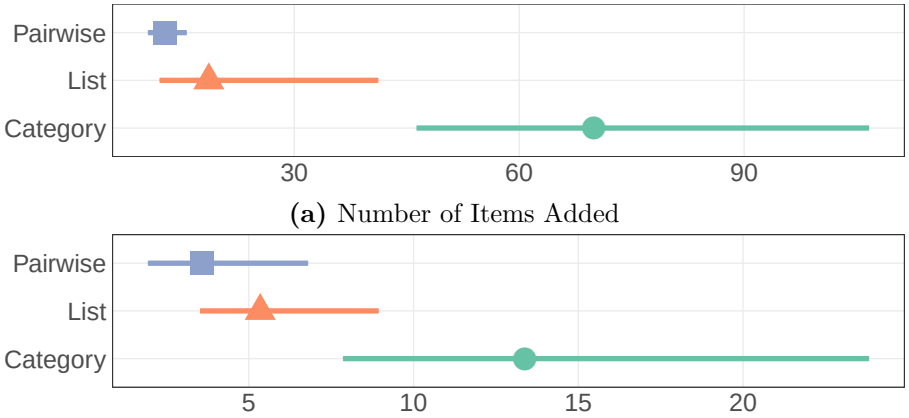
In addition, self-reported quantitative measures were collected using the post-test survey. Finally, using free-response questions, we also collect participant comments on their ranking strategy and experience using the college ranker.

3.2 Results

144 participants were recruited through Amazon's Mechanical Turk for the study. Out of the total, 49 participants were randomly assigned to the Sub-list preference collection mode, 45 participants to the Categorical Binning mode, and 50 participants to the Pair preference mode. For each measure we compute quantitative results comparing each study condition. In response to concerns about the limitations of null hypothesis significance testing [47, 149], we model our analyses on HCI research that seeks to move

[1] We note this is a distinct measure from the total number of interactions performed by the user since they may add, remove, and swap many items during the build phase before ranking.

beyond these limitations (e.g. Dragicevic [57]). Following Cumming [47], we compute 95% confidence intervals using the bootstrap method, and use Cohen's d to measure effect sizes (the difference in means of the conditions divided by the pooled standard deviation). Error bars in figures are the 95% confidence intervals (CIs).



(a) Number of Items Added

(b) Number of Items Removed.

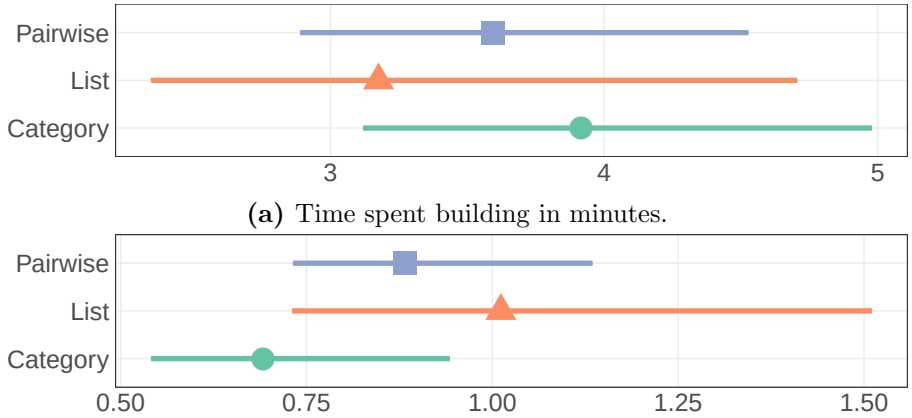
Figure 3.3: Comparing the number of user interactions across preference collection modes.

3.2.1 Elicitation Techniques and Observed User Behavior

Effect on Number of Interactions We found that the average participant who was assigned to the Categorical Binning mode interacted with significantly more items (M = 69.9 items added 95% CI [46.4, 106.6]) from the dataset than those participants assigned to the Sub-list or Pair modes (M = 18.6 items added 95% CI [12.1, 41.1], and M = 12.7 items added 95% CI [10.5, 15.6] respectively). Results are shown in Figure 3.3. We interpret the confidence intervals following Cumming's methodology [47]. Given the upper and lower limits of the confidence intervals, the average participant in the Categorical Binning group added at least 5 additional items during the build phase as in the other two conditions, and up to 95 items more. The effect size as measured by Cohen's d between Categorical Binning and Pair preference modes is large: $d = 0.79$ $[0.57, 1.06]$ and between Categorical Binning and Sub-list modes $d = 0.66$ $[0.27, 0.96]$. There is a small effect observed between Sub-list and Pair preference modes $d = 0.22$ $[-0.24, 0.5]$.

We also count number of items removed from the preference collection interface during the build phase. While there are fewer remove interactions on average, we observe a a similar effect across modes. The average user assigned to Categorical Binning mode removed more items (M = 13.4 items removed 95% CI [7.9, 23.8]) than in Sub-list mode

(M = 5.3 items removed 95% CI [3.5, 8.9] or Pair modes (M = 3.6 items removed 95% CI [1.9, 6.7]).The effect between Categorical Binning and Pair preference modes has effect size $d = 0.51$ $[0.23, 0.73]$, between Categorical Binning and Sub-list $d = 0.42$ $[0.11, 0.71]$, and between Sub-list and Pair modes $d = 0.28$ $[-0.37, 0.78]$.



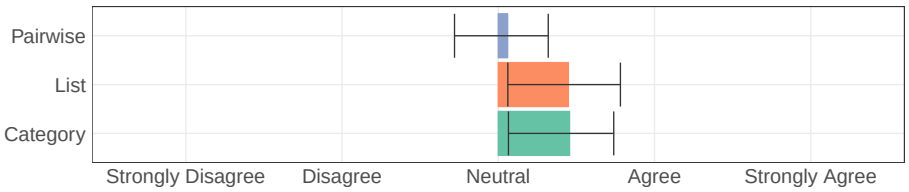
(a) Time spent building in minutes.

(b) Time spent exploring in minutes.

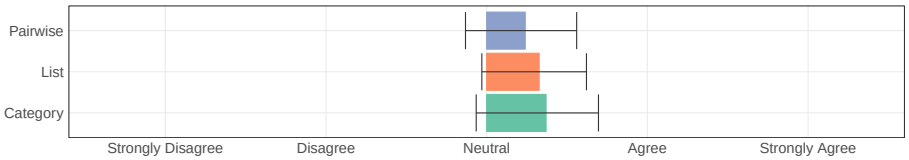
Figure 3.4: Comparing the time spent interacting with the preference collection interface during the build phase.

Effect on Time Spent Interacting Despite the significant difference in the number of items added, we do not see a corresponding difference in the amount of time spent entering preferences in the rank building phase (Fig. 3.4a). Sub-list time building (M = 3.2 minutes 95% CI [2.3, 4.7]), Categorical Binning time building (M = 3.9 minutes 95% CI [3.0, 5.2]), and Pair time building (M = 3.6 minutes 95% CI [2.8, 4.6]) do not exhibit any significant effect from the preference collection mode used. Rank exploration time is not significantly impacted by preference elicitation technique either (Fig 3.4b). Results show similar Sub-list time exploring (M = 1.0 minutes 95% CI [0.7, 1.4]), Categorical Binning time exploring(M = 0.7 minutes 95% CI [0.5, 0.9]), and Pairwise time exploring (M = 0.9 minutes 95% CI [0.7, 1.1]).

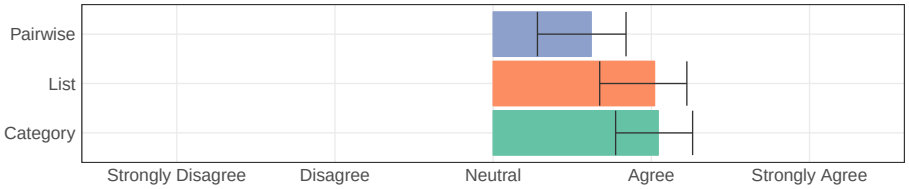
Effect on User Satisfaction. We include some examples of the qualitative statements presented to users in the Post-study survey (Fig. 3.5), which did not show significant differences between preference collection modes. Even though users didn't report a difference in the use of these three modes, the difference in the number of interactions tells a different story.



(a) The final ranking generated by the system reflected my personal preferences about colleges.



(b) I would use this system to make decisions about colleges.



(c) When I was entering colleges into the build tool, I could easily express my preferences.

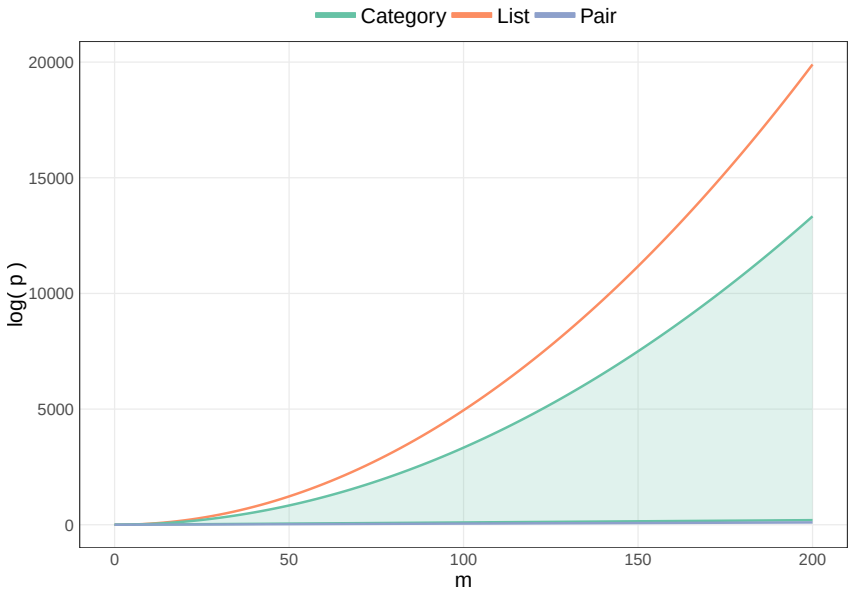
Figure 3.5: Qualitative assessment. Users were asked to indicate their level of agreement with each statement.

3.2.2 Elicitation Techniques and ML Implications

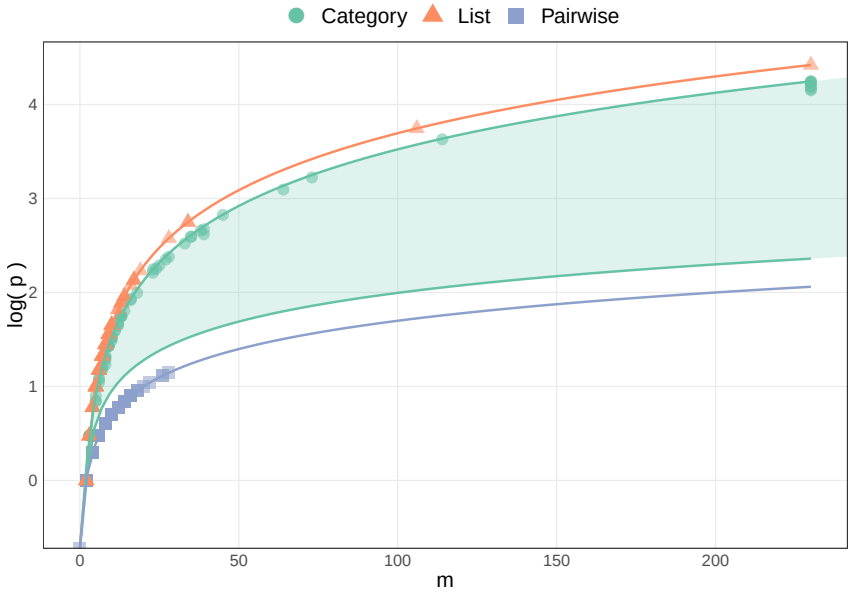
Effect on Pair Growth Rate. As discussed in detail in 3.1.2, to learn a global ranking over the dataset the ranking engine is trained over object pairs. To evaluate the effect of alternative preference collection methods on system performance, we consider the number of pairs that can be extracted using each elicitation technique. We derive the *pair growth rate* for each mode (shown in Fig. 3.6a) which captures the number of pairs n that is generated given m data items entered into the preference collection interface. Here we consider the number of items collected when the user clicks "Rank!" to generate the global ranking. Since objects are arranged differently in each preference collection mode, there is a different pair growth rate associated with each.

Sub-list Ranking Pair Growth Rate: The list view specifies an explicit order over the set of items. The pair growth rate of the list is thus $\binom{m}{2} = m(m-1)/2$, reflecting all possible ways of choosing ordered pairs from the list. This is a quadratic growth rate, meaning if m items have been added into the list, then the number of pairs implicitly specified is on the order of $O(m^2)$.

Categorical Binning Pair Growth Rate: The categorical comparison mode is



(a) Growth rate curves. Includes min and max rates for Categorical Binning, with the shaded region covering possible rates due to an unequal distribution of items in each bin.



(b) Actual number of pairs generated shown over growth rate curves, shown in log scale.

Figure 3.6: Pair growth rates comparing m the number of items in the preference collection interface to against p the number of pairs extracted.

more difficult to quantify, since a different number of objects can be added to each of the 3 bins. In the worst case, $m-1$ items will be placed in one bin, and only one item placed in a second bin. In this case, only $m-1$ pairs would be formed ($O(m)$ linear growth

rate). However, assuming an equal distribution of $m/3$ objects in each bin, many more pairs are formed between the bins. In this best case, the growth rate is $\frac{m^2}{3}$ possible pairs. So, while this method is also quadratic in the best case, it has a slower minimum growth rate than the list comparison. The max and min rates are both shown in 3.6a, with the shaded region covering the possible range of pairs resulting from an uneven distribution of items across bins.

Pair Preference Pair Growth Rate: This mode is most directly aligned with the pairwise formulation of the underlying ranking algorithm. Here the user is asked to specify each pair explicitly, meaning this is the most labor-intensive of the three modes. Since two items are required to form every pair, the growth rate is $m/2$, also linear and even slower than the min rate of the categorical mode. Fig. 3.6b shows the actual number of pairs generated by users laid over the pair growth rates, shown in log scale for readability. We can see that while the number of pairs generated by the Sub-list and Pairwise modes are fixed dependent on m, for the Categorical mode some values fall in the shaded region. On the whole it can be observed from this chart that users tend to distribute data evenly among the bins in practice, yielding pair numbers of close to the maximum rate for the Categorical mode.

Effect on Training Data Size We found that the average participant who was assigned to the Categorical Binning mode generated significantly more training data pairs (M = 2185 pairs 95% CI [1170,3805.5]) than participants assigned to the Pairwise mode (M=4.5 pairs 95% CI [3.9,5.2]), as indicated by Cohen's d with effect size d=0.63 [0.45,0.86]. Sub-list mode also resulted in fewer pairs on average (M = M=401.4 pairs 95% CI [55.1,1875.2]). These results are shown in 3.7. The effect size as measured by Cohen's d between Categorical Binning and Sub-list modes d=0.44 [0.01,0.69], and between Sub-list and Pair preference modes 0.2 [0.16,0.26].

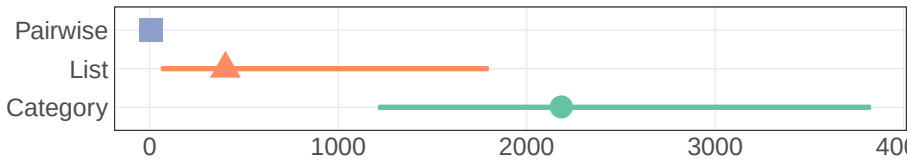


Figure 3.7: Number of pairs generated from user preferences.

3.3 Discussion

The results of our study suggest that mode of preference collection can significantly influence the number of interactions performed by users (Fig. 3.3), as well as the amount of training data provided to the ranking engine (Fig. 3.7), without impacting the amount of time spent by users (Figure 3.4), or ease of use of the ranking tool (Fig. 3.5). One general implication of these results is that the Categorical Binning mode provides the best tradeoff between user effort and training dataset size. We next turn our attention to other possible explanations for these findings, and implications for the design of interactive ranking systems.

3.3.1 Categorical Binning: High User Engagement and Expressiveness?

In the Categorical Binning mode, users interact with a large amount of data quickly, organizing information using broad strokes, and providing the most training data to the ranking engine on average. An added benefit of this interaction mode is that categories can capture more ambiguity on the part of the user, in comparison to the Sub-list and Pair modes. For example, items placed in the top category may be perceived by the user to be preferred to the other items in the dataset, however, people are not forced to impose a strict order among them. Future work might investigate more closely the possible variations in user behavior and intent *within* binning modes, as it is possible that a person would want to organize both between (the current focus) as well as within categories.

Interaction modes using Pair preferences or Sub-list ranking resulted in smaller input from users. Sub-list collection mode has the fastest pair growth rate, however in practice many fewer training pairs were generated using this mode. Lists have high potential, but rarely do people use them to their full capacity. Future research could draw on work in human-computer interaction targeting search elicitation strategies, such as work from Agapie *et al.* which explored UI components that led people to longer search queries [1]. Merging "nudging" threads of research with rank preference elicitation may yield additional evidence-driven design guidelines that better optimize the relationship between the user and the underlying ranking algorithms being explored in similar systems today [72, 104, 105, 147].

The Pair preference format received the lowest rating from study participants. This aligns with the fact that the slow pair growth rate means that much more user effort is

required to enter enough data to learn a useful ranking.

3.3.2 Towards Compositions of User Elicitation Techniques

Although the results of this experiment indicate that users add significantly more data with the Categorical Binning technique, it should not be taken to mean that categorical techniques are strictly superior to other elicitation possibilities. We posit that compositional approaches to user preference elicitation may be a path towards mitigating the drawbacks of each approach while maximizing the amount and quality of information the user provides to the system. For example, given that the Sub-List mode has the fastest growth rate, and the fact that some participants (outliers) were observed to use the list technique to its full potential, future interfaces could possibly combine the benefits of both preference collection modes. Exploring these possibilities will likely require additional experiments, and possibly the creation and evaluation of novel elicitation interaction techniques. Recent work from Wall *et al.* on the Podium system can be taken as a point in this design space [147], given their system allows users to directly manipulate a ranking results list. A recently proposed approach of "blended recommendation" [112] which explores the use of manual interactions such as data attribute filtering and weight adjustment combined with automated recommendation could also inform design for interactive ranking.

4

RanKit System for Personalized Decision Support

Insights gained from our user study on preference collection methods and experience designing the interactive College Ranker tool inform the design of RanKit, a general purpose mixed-initiative system for interactive ranking. RanKit allows users to leverage their intuition or partial understanding of a complex dataset to extract a global ranking model. Careful visual interface design avoids biasing the user to any pre-specified order of items, ensuring the resulting ranking is driven purely from user preferences. The system not only learns relationships among the data points automatically, but it does so in a way that is transparent to the user. Visual feedback communicates the impact of user interactions in real time, allowing the user to drive the ranking process by changing their input or specifying additional preferences. We demonstrated the RanKit system at the 2018 CIKM Conference [104].

> Caitlin Kuhlman, MaryAnn VanValkenburg, Diana Doherty, Malika Nurbekova, Goutham Deva, Zarni Phyo, Elke Rundensteiner, Lane Harrison. 2018. Preference-driven Interactive Ranking System for Personalized Decision Support. *In The 27th ACM International Conference on Information and Knowledge Management, ACM 2018*

4.1 RanKit System Overview

The RanKit system overview is depicted in Figure 4.1. Datasets to be analyzed are imported and preprocessed to clean missing values, encode categorical attributes, and normalize the data, before being housed in the RanKit Data Store. Plug-and-play design

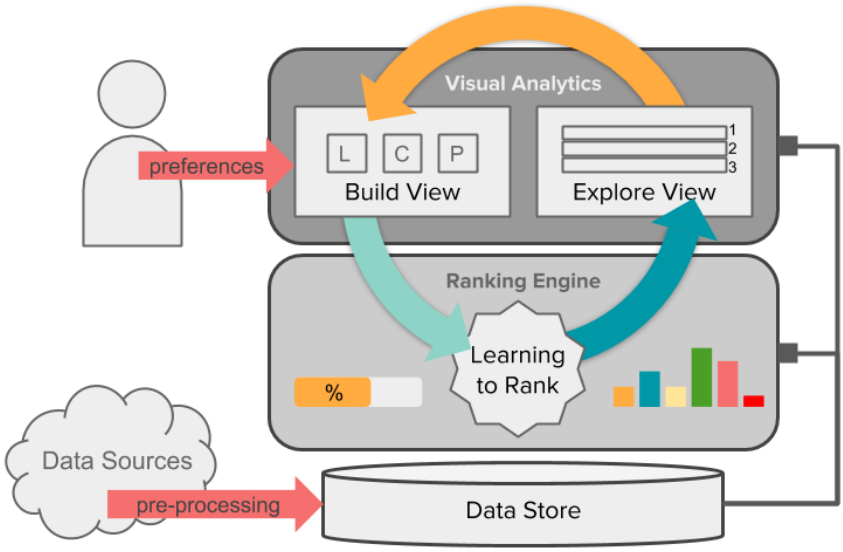


Figure 4.1: RanKit system overview.

allows for any pairwise learning-to-rank algorithm to power the backend Ranking Engine. User preferences are collected in the Visual Analytics layer (top of figure). From these entries, pairwise relationships between items are extracted and sent to the machine learning algorithm at the Execution layer (middle of figure). RanKit computes the ranking model in real time. The weights of data attributes are updated incrementally with each change in the build view. In addition, diagnostic metrics are continually computed and sent back to the analytics layer. For our demonstration, we allow the user to choose among the three preference collection modes evaluated in our user study [103]: *Sub-list Ranking*, *Categorical Binning*, and *Pair* preference modes. At any time, the user can switch to the Explore view, triggering ranking predictions to be made and the global ranking over all items to be sent to the front end for display. The user is free to alternate between visual modes to refine their ranking.

4.2 RanKit Key Innovations

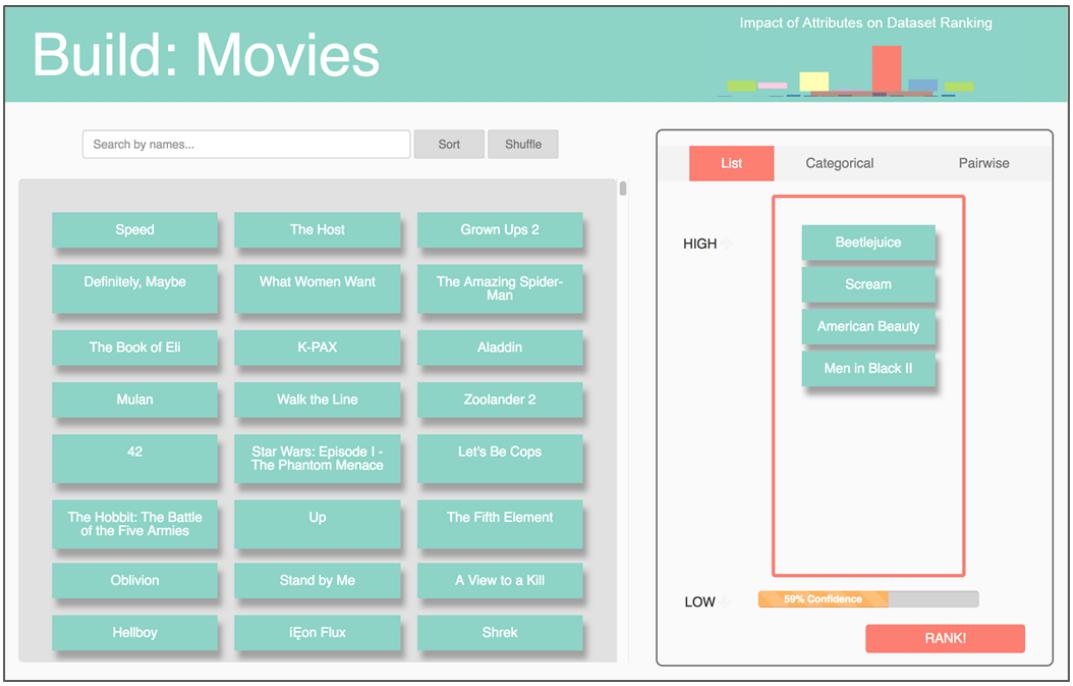
Build, Explore, Explain Paradigm. The design of the RanKit system addresses a number of the considerations detailed in the requirements analysis given in Chapter **??** using a "Build, Explore, Explain" approach. RanKit features dual interfaces that separate out the two core interaction modes, namely, the "Build" view for specifying preferences over items and the "Explore" view for evaluating data attributes. This separation avoids predisposing the user toward any particular preference choices (as recommended

in requirement **R2**). These views are shown in Figure 4.2.

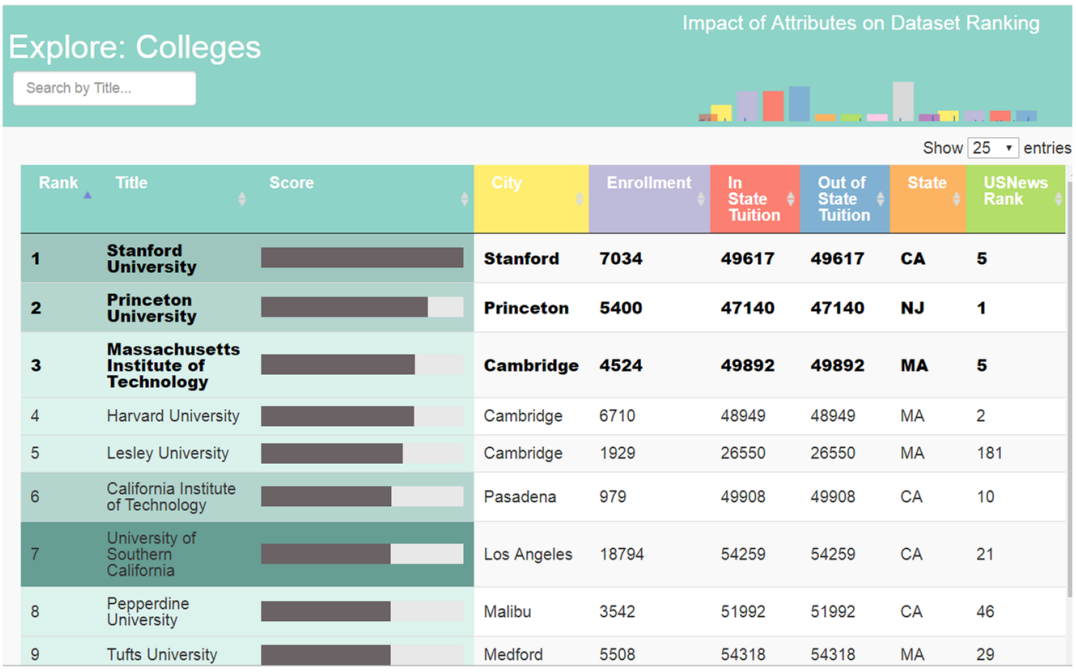
Build View. The Build view (Fig. 4.2a) has two main components - the data is displayed on the left side of the screen and the preference collection interface on the right. To avoid biasing the user toward any pre-ranked numeric or lexicographic ordering [131], we display the colleges from our target dataset in a "data pool" where they are represented only by name, arranged in a grid format. As a user may want to further learn about each college and its properties, we provide the attribute values for each item in a tooltip accessible on hover. On page load, the dataset is randomly shuffled and displayed in the data pool. Multiple navigation modes are offered: users can search for a college by name, sort the data alphabetically, or use the"Shuffle" button to randomly permute the data. All three build modes employ the same basic interaction – to enter their preferences, the user drags colleges from the data pool into the preference collection interface (3.1) on the right. Users can move as many objects as they want, swapping their order and moving them between the pool and the different fields within the comparison tool.

Explore: Global Ranking Interface. Once the user hits the "Rank!" button, they are redirected to the "Explore" view. A thinking step displays a spinner and message "we are computing your global ranking ... " to communicate to the user the conceptual division between the data they have manipulated to train the underlying model, and the learned global ranking displayed in the explore view. The Explore view (Fig. 4.2b) visualizes the learned college ranking in a table. To easily identify and evaluate their input from the previous view, colleges that the user manipulated are highlighted with bold text. Here users can evaluate the relationship between the ranking and underlying data attributes, along with the learned expression of the ranking as a weighted data attribute model. Iterating between these two interfaces, users can continually refine their ranking and drive the knowledge generation process through collaboration with the learning method to achieve a desired result.

Explain. To further elucidate the rank learning process, we incorporate "Explain" features throughout the RanKit system, aiming to build user trust by explicitly communicating the amount of uncertainty around the ranking currently learned by the system (requirement **R4**). This is realized by showing incremental changes to the ranking model resulting from user interactions. In addition, we visually encode the confidence of result-



(a) The RanKit Build view consists of the data pool on the left and preference collection interface on the right. Preferences over the Movies dataset are shown in Sub-list mode.



Rank	Title	Score	City	Enrollment	In State Tuition	Out of State Tuition	State	USNews Rank
1	**Stanford University**		**Stanford**	**7034**	**49617**	**49617**	**CA**	**5**
2	**Princeton University**		**Princeton**	**5400**	**47140**	**47140**	**NJ**	**1**
3	**Massachusetts Institute of Technology**		**Cambridge**	**4524**	**49892**	**49892**	**MA**	**5**
4	Harvard University		Cambridge	6710	48949	48949	MA	2
5	Lesley University		Cambridge	1929	26550	26550	MA	181
6	California Institute of Technology		Pasadena	979	49908	49908	CA	10
7	University of Southern California		Los Angeles	18794	54259	54259	CA	21
8	Pepperdine University		Malibu	3542	51992	51992	CA	46
9	Tufts University		Medford	5508	54318	54318	MA	29

(b) RanKit Explore view shows a ranking learned over the Colleges dataset.

Figure 4.2: Views in the RanKit system.

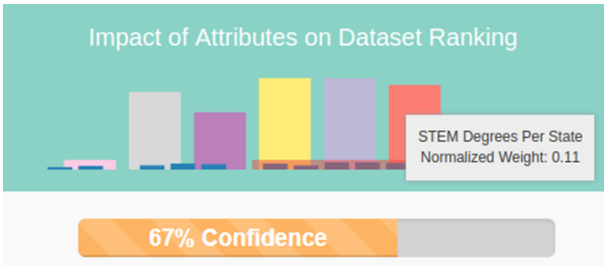


Figure 4.3: RanKit Explain features: (top) Normalized weighting for each attribute in model. (bottom) progress bar shows overall confidence score.

ing predictions and the quality of the overall model. We design our uncertainty criteria by closely studying active learning techniques for ranking [131], and informed by our complexity analysis.

Measuring uncertainty to foster user trust and understanding To measure the user's progress and evaluate the quality of the ranking model we design confidence metrics which are continually updated and visualized for the user. In turn, the resulting global ranking will match their expectation, and produce a model they trust. To evaluate the overall expected quality of the ranking model, we measure the number of concordant and discordant pairs predicted for the training dataset, adjusted by an estimate of a sufficient minimum number of training pairs. This confidence score is displayed as a progress bar in the Build view, instantly communicating to the user that as they add more data, the quality of the model improves. This encourages interaction in this crucial build stage to ensure that the resulting model will be able to distinguish between items in the dataset in a meaningful way.

In addition to measuring overall model quality, we design a metric to evaluate the predictive ability of the model for each individual item in the dataset. In RankSVM, the most ambiguous pairs of items are those closest to the decision boundary. Therefore, to derive a score for each item, we aggregate the distances to the boundary over all pairs it appears in. If the pairs containing a particular item tend to be far from the decision bound, then it will have a high confidence value. If many pairs containing the item are close to the decision boundary, this means its rank is difficult to distinguish from many other items. Therefore the confidence of the prediction will be low. Individual item confidence scoring allows the user to identify items which may be useful to enter in the build view to provide more information to the ranking engine.

4.3 Case Study Evaluation

In our demonstration of RanKit we included several real datasets from diverse domains for analysis, including colleges [1][2], movies [3], games [4][5], sports [6], and the US economy [7]. The following case study on college ranking illustrates the capabilities of RanKit.

Choosing a dataset and comparison method. Alice is a high school junior deciding where to apply for college. She has toured several schools around her home state. Some made a good impression while others were definitely not for her. She also has a couple of "dream" schools that she would love to attend. Given this partial understanding of a few colleges, Alice would like to know how her other potential choices stack up. She uses RanKit to analyze the colleges dataset containing both numeric and categorical attributes of US colleges including the size, cost of tuition, geographic region, and acceptance rate. Alice selects the colleges dataset to load and display the cleaned data in the Build view.

Entering preferences in the Build tool. Figure 3.1a shows the Build view layout. Colleges to be ranked are displayed on the left-hand side of the screen in a grid, with items arranged randomly. This avoids implying preference according to numeric or lexicographic order. Data attributes for each college can be accessed as a tooltip.

On the right side of the Build view, Alice has a choice of three preference collection formats. Alice decides that Categorical Comparison mode is the easiest way to enter her preferences among the colleges she is familiar with. She uses the search box to find her "dream colleges" in the data pool and drags them to the high category. She puts the schools she dislikes in the low category. She can move as many items as desired, swapping their order and moving them between the pool and the different fields within the comparison tool.

Once Alice enters items into two different categories, reactive visualizations begin showing the incremental progress of the ranking engine. The model weights appear as a bar graph in the header of the page, changing in real time to show the impact of item preferences on the learned importance of the attributes. In addition, an overall confidence

[1] https://collegescorecard.ed.gov/
[2] https://www.usnews.com/best-colleges/rankings/national-universities
[3] https://www.kaggle.com/rounakbanik/the-movies-dataset
[4] https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings
[5] https://www.kaggle.com/mrpantherson/board-game-data
[6] https://www.sports-reference.com/cfb/
[7] http://matters.mhtc.org/

score for the model is shown in a progress bar below the comparison tool. At this point, the progress bar shows a low confidence score meaning that no reliable rankings can be generated using her current input preferences.

Transitioning from the Build view to the Explore view. Impatient to see results, Alice hits the "Rank" button and is redirected to the "Explore" view. The global ranking over all colleges is displayed in a table, along with data attribute values. Several visual encodings communicate information learned from the interactions in the previous view. Color is used to visually associate the column headers with the attribute weights in the bar graph at the top of the page. Items entered in the Build view are highlighted with bold text. The overall score assigned to each item is indicated with gray horizontal bars (the numeric score is available on hover). These scores are determined by a weighted combination of the data attributes according to the learned model. In addition, a confidence value is determined for each item. This is visualized using the background coloring of the first columns of the table, with darker colors indicating higher confidence (the exact confidence score can be accessed as a tooltip). At a glance, Alice finds that none of the colleges listed at the top of the ranking have high confidence values. This indicates that she has not yet entered sufficient information. Alice thus decides to return to the Build view to improve her model.

Using feedback to improve ranking. This time, Alice adds the colleges that she visited and liked into the medium category. She notices that the confidence value in the progress bar goes up. Alice uses the "Shuffle" button to permute the items in the data pool to browse the rest of the colleges. Reflecting on what her goals are for college, Alice decides to add colleges with strong Science programs to the High category, and some without to the Low category. Continuing this way, Alice drags colleges into the comparison interface until the progress bar reaches 99%. Switching back to the Explore view, Alice sees the ranking shown in Figure 4.2b.

Understanding the ranking model. This time, Alice is pleased to see several of her favorite colleges are at the top of the ranking, as well as some she hadn't considered. The confidence values for these individual items are higher now. Alice hovers over the bar graph in the top right corner to see which attributes contribute the most to her model. She sees that enrollment is a significant contributor to the ranking, and observes that the colleges she prefers have a tendency to be large research universities.

Making a decision from the data. Looking at her ranking, two colleges that are unfamiliar to Alice stand out. California Institute of Technology and University of Southern California are ranked among the top ten colleges with high confidence values. Alice decides to spend additional time researching these schools to better inform her college application choice.

5

Related Work

In recent years, several multi-attribute ranking systems have been developed to help users visualize and interact with rankings [29, 72, 104, 105, 126, 140, 147]. Manual systems focus has on aiding users in *adjusting data attribute weights* of a multi-criteria ranking and visualizing the resulting impact across attribute subsets [126], alternative rankings of the same items [72], and rankings over time [140]. In mixed-initiative systems [86] including Podium [147] and our RanKit system [105], a machine learning algorithm learns a global ranking of dataset based on the user's *preferences over a subset of items.* these systems aim to better capture the user's intuitive understanding of the relative value of the objects to be ranked.

5.1 ML Algorithms for Interactive Learning-to-Rank

Mixed initiative ranking systems employ learning-to-rank algorithms, originally developed and most commonly applied for Information Retrieval [111] and Recommender Systems [2]. Existing systems [104, 105, 147] adopt a "pairwise" formulation of the learning-to-rank problem [111]. First introduced by Herbrich in [83] for ordinal regression, and later applied to document retrieval by Joachims [91], the pairwise formulation allows a ranking to be learned using a binary classifier applied to pairs of data instances. Building on this result, any classification model can be employed for learning-to-rank, and many have been proposed [27, 67, 91, 125]. As detailed in Section 3.1.2, RankSVM [91] features a number of properties that naturally fit the interactive ranking task.

Ranking in mixed initiative systems [104, 147] is semi-supervised, where a subset of data is manipulated by the user, and labels extracted from these interactions are leveraged to transform the entire dataset into a consistent ranking. Ideally, methods

employed for this task will be effective at rank generation while requiring only a minimal effort on the part of the user. Much recent work has been done to characterize the sample complexity of pairwise formulations of ranking problems [64, 132, 151]. In particular, Wauthier et al. [151] show that $O(n)$ training pairs are required to learn a model that will perform better than random in expectation. However, for our purpose, labels provided by users are not selected randomly as assumed in this analysis. Additionally, in these cases, a true underlying model is not necessarily assumed which can be learned from the data attributes. For training over data attributes, active learning methods [42] find the most informative pairs to label, again improving the sample complexity of the problem [114, 131, 158]. Thus active learning techniques have potential to improve the amount of effort required by the user to learn high quality rankings.

5.2 Preference Elicitation Techniques for Interactive Ranking

Interactive ranking systems use different mechanisms to collect user preferences. In an initial prototype [104], we demonstrated interactive ranking in the context of measuring economic competitiveness. There users directly specified pairs where one object is preferred to another. In Podium [147], a semantic interaction approach [62] is applied. Preferences are inferred from users' interactions re-ordering items in a list, rather than being directly specified by the user. For insight into the problem of collecting user preferences, we can also look to the wealth of research around HCI for recommender systems [28, 82, 99, 100], which rely on ranking according to user preferences as a subtask. However, it is important to note some key differences between recommendation and ranking systems. Recommender systems aim to automatically find interesting items in a dataset, while interactive ranking systems help users find patterns in the data through exploration. For recommendation, user preferences are often collected *implicitly*, based on interactions such as search queries or click-through logs. Preferences of *multiple users* are typically aggregated, using collaborative filtering [2] to make predictions. In an interactive ranking setting, a subset of data is manipulated by a *single user* in order to deliberately train a ranking model. The model is then applied to create a global ordering over the same dataset in a semi-supervised [144] manner.

For recommendation systems, it has been observed that user satisfaction is positively impacted by a sense of control over the recommendation process [28, 100]. As discussed

in a survey by He et al. [82], explicit interaction and visualization have been incorporated into a number of recommender systems to improve qualitative aspects of the recommendation process. User preferences may be used to match similar users or address "cold start" problems. The most prevalent way recommender systems collect information about items is to have users rate them (such as giving a rating out of 5 stars for instance). However, it has been shown that such absolute evaluations are difficult for users to perform, and studies demonstrate that user ratings can be inconsistent or inaccurate [7] Humans are more cognitively adept at making *relative judgments* [30, 38]. Some interactive recommender systems accomplish this by allowing users to group together items they consider similar [82, 112]. Organizing recommendation results in categories has also been shown to help users identify qualities such as diversity [88]. One recommender system [113] evaluated the impact of collecting pairwise preferences over a subset of items, finding that it improved user satisfaction.

5.3 Visualizing Ranked Data

A common approach to visually represent rankings is in a table or list view, which can be intuitive to navigate given most users' familiarity with tools such as spreadsheets. Sophisticated designs have added interactivity and additional functionality to such ranked data views. As an example, the Lineup system [72] accomplishes a number of things in one integrated table view. Items are arranged in ranked order with the underlying data attributes indicated using color. Attribute impact on the score for each item is indicated by the size of horizontal bars. Stacking these bars gives a visual indication of the distribution of the scores through the list. Views also compare multiple rankings against one another. One thing that is hard to convey using this table approach however is a *holistic* understanding of the ranking. Users have to scroll through the table to see all the results. The Lineup and Podium systems overcome this using a "snapshot" views that summarizes information concisely in a small histogram.

Stacked bar charts are popular for comparing attribute weightings [29, 72, 126, 147]. In the Valuecharts system [29], three visualizations are suggested for the comparison and manipulation of attribute weights. A stacked bar chart shows the values of weighted attributes for each item being ranked. Proportional bar charts represent each attribute in a separate chart which is scaled according to the weight of the attribute, to help the user compare the values of attributes across objects. Last, an exploded divided bar chart gives users the ability to manipulate the weights of attributes to understand the

sensitivity of the model to the attributes.

Interaction with these visualizations facilitates exploration of different possible weight combinations. However, assigning attribute weights is a taxing process. The user may have to guess at initial weightings, and the effort required to explore the entire space of possible combinations can be prohibitive. One solution is to visually represent a space of possible rankings. Many approaches have been proposed for this task. In the study of permutations, graph representations (i.e. Cayley diagrams) and permutation polytopes [156] have been used to represent the groups of permutations visually. However, such tools are limited to rankings over small sets of items. For high-dimensional rankings, Kidwell et al. [95] use the Kendall tau distance to implement multidimensional scaling (MDS) over incomplete rankings to visualize them in two dimensional space. One system, WeightLifter [126], automatically characterizes relevant regions of the ranking space to provided guidance on the choice of weights to the user. For instance, stable regions of rank sensitivity are visualized to help the user understand how small changes in the data affect the rank outcome. Many aspects of interactive visualization of multiple rankings remain to be explored. We discuss possible research directions in Chapter 12.

Part II

Ranking for Fair Decision Making

6

Introduction: Fair Ranking

6.1 Motivation

As sophisticated machine learning increasingly impacts our lives on and offline, there is growing concern that discriminatory practices will be baked into automated decision models [12, 124]. Research on algorithmic fairness aims to ensure fair practices with respect to sensitive data attributes, e.g., race, gender, or age, which by law are not permitted to determine decision outcomes. The bulk of recent work in this area [39, 44, 59, 65, 80, 98, 130, 160] has targeted classification tasks, where predictive models are used to determine a binary outcome. A number of fairness criteria have been proposed for this task, and the benefits and trade-offs between criteria have been explored [44, 130]. It has been shown that in general, not all criteria can be simultaneously satisfied [39, 98]. The correct fairness criteria to apply is therefore highly dependent on the problem domain. Recently attention has shifted to include *fair ranking* which is critically important for information retrieval (IR) tasks underlying socio-technical systems. The need for meta-analysis of fair ranking evaluation metrics has been observed by leading fairness researchers [123], however to-date this has been an understudied area.

Increasingly, rankings not only mediate people's access to information online, but also screen and filter candidates for tasks such as hiring and university admissions [37, 70, 78, 150]. The fairness of ranked search results may be impacted by many factors, including historic bias or misrepresentation of groups in training data [141], bias encoded in tools used to parse the data such as for image [26] and text analysis [21], as well as implicit bias inherent in users' interaction behavior [33]. Ranking algorithms used in IR systems may exacerbate such unfairness in a rich-get-richer fashion [123].

IR systems often serve multiple stakeholders and optimize for multiple concurrent

goals (e.g. relevance, novelty, etc.), and fairness definitions are highly dependent on context. Therefore in this dynamic setting multiple fairness metrics are required as appropriate for various scenarios. An in-depth understanding of proposed metrics is required to guide practitioners in choosing the right metric for their application, and to facilitate oversight and agreed upon standards for measuring unfairness in rankings.

6.2 State-of-the-Art Fairness Metrics for Ranking.

Initial proposed fairness definitions for ranking mainly target group fairness [34, 70, 106, 141, 155, 161, 162], which aims to ensure equal treatment or outcomes for groups of people according to protected data attributes such as race, gender or age. The majority of these works adopt statistical parity measures. Statistical parity is one of the simplest fairness definitions, which dictates that each group receive fair proportions of favorable outcomes. Statistical parity is particularly useful when there is a diversity requirement in place to achieve distributional justice for groups that have historically been discriminated against (e.g., motivating the 80% rule in discrimination law [65]).

However, it has been observed that enforcing statistical parity may exact a high toll in terms of predictive accuracy, and possibly infringe on fairness for individuals [58]. For such reasons, adaptations of other fairness definitions from classification have also been proposed for rankings. Individual fairness originally for classification proposed by Dwork *et al.* stipulates that similar individuals should be treated in the same way. This standard is applied for rankings by Biega *et al.* [18]. Equalized Odds criteria, proposed for classification by Hardt *et al.* [80], seek to ensure that the probability of an object being assigned a particular label by the classifier is independent of its group membership, conditional on the true class label. To verify this, Equalized Odds stipulates that the *false positive and true positive error rates* must be similar across all groups. Fairness based on equal error-rates for groups is proposed in our work (presented in the next chapter [106]). Other analogues have been proposed for IR by Singh and Joachims [141] by considering exposure of items in proportion to relevance. Finally, causal definitions of fairness [119, 152] aim to understand the relationships between data attributes and predicted outcomes, as an alternative to measuring fairness using evaluation metrics.

Limitations of Metric Design. Fairness definitions for classification tasks hinge on the fact that some people being evaluated will receive a favorable outcome and some will not, corresponding to a positive class and a negative class. For ranking tasks determining

a preferred outcome is more subtle. In this case rank position determines the outcome for the items being ranked, with distinct advantages conferred to those items at the top. However in rankings *position is relative* – it depends on many factors such as the quality of the rest of the items in the list and the importance of specific positions in the ranking (i.e. position bias [92]). Proposed fairness metrics for ranking attempt to account for this by measuring *group advantage* in a ranking following established strategies in IR: employing top-k analysis [34, 155, 161], pairwise inversions [106, 121], and cumulative discounted metrics [70, 141]. Notions of user attention [18] and exposure of items being ranked [141, 162] have also been used to frame the problem.

Comprehensive comparative analysis of these approaches is lacking. It remains an open question whether the advantage being measured by one metric is actually the same phenomena being measured by another using a different formulation. Further, to date guidance is unavailable for deciding when say, a pairwise metric might be preferred to an exposure-based metric or a top-k metric, or when they are they all equivalent. In fact, it is not clear what evaluation strategy should be used to even compare metrics that quantify group advantage in different ways. If this basic premise is not well understood, then designing even the simplest fair ranking metrics is fraught with uncertainty.

Therefore in part two of this dissertation we investigate fair ranking evaluation metrics in depth. In Chapter 7 we propose a pairwise formulation of fair ranking evaluation metrics which allows error-based notions of fairness to be adapted from classification to ranking. We use these metrics to demonstrate tradeoffs between different notions of fairness. Then in Chapter 8 we dig deeper into a single definition of fairness – statistical parity – and compare ways that group advantage can be compared in rankings. Finally we consider fair ranking in a new problem setting of rank aggregation in Chapter 9. First we formally define the fair ranking problem.

6.3 Fair Ranking Problem Formulation

Ranking can have different meanings in different contexts, and ranking models can be trained over various types of ground truth information. Rank predictions can be learned from training data with binary labels (e.g., in bipartite ranking [45]) or discrete labels with ordered classes (i.e., ordinal regression [83] with labels such as “best”, “neutral”, “worst”). Traditional regression ranks according to continuous scoring functions. Learning-to-rank approaches also include pairwise and listwise models [111].

Therefore, to be widely applicable, we target general rankings with a model-agnostic

approach. We assume only that an ordering is given for a set of *candidates* $x_i \in X$. This determines a *ranking* of X which is a permutation $\rho = [x_1 \prec x_2 \prec ... \prec x_n]$ over all candidates. Here $\prec$ is a complete ordering relation on X such that $x_i \prec_\rho x_j$ implies that x_i appears at a *more* preferred position than x_j in the ranking ρ. The position of a single candidate x_i in the ranking ρ is denoted $\rho(x_i)$. We adopt the convention that low number positions are favored over higher ones, i.e. $\rho(x_i) = 1$ is the best rank position.

Unique to the context of fairness analysis, each candidate being ranked also has associated *protected attributes* (e.g., race, gender, or age). These attributes partition the dataset into two or more disjoint or overlapping *groups* $\{G_1, \ldots, G_m \mid \cup_{i=1}^{m} G_i = X\}$. Traditionally, one group corresponds to minority or otherwise disadvantaged groups of people according to legally protected data attributes such as race, gender, or age. Fairness of a predicted ranking $\hat{\rho}$ is assessed according to some *Fairness Criteria* which relies on a group error function L.

Definition 6.1. *Given a group error metric $L_{G_i}(\rho, \hat{\rho})$, a **Fairness Criteria** (FC) is an evaluation rule which designates a ranking $\hat{\rho}$ as fair in relation to a true ranking ρ if:*

$$L_{G_i}(\rho, \hat{\rho}) \cong L_{G_j}(\rho, \hat{\rho}) \ , \ \forall G_i, G_j \quad i \neq j$$

Fairness is evaluated by checking whether the error for each group is similar, or within some threshold, indicated by the symbol $\cong$. The larger the difference in the errors for each group, the more unfair the ranking is considered to be. Our assessment therefore hinges on the choice of a rank-appropriate group error function L.

6.4 Defining Groups

Although most work on algorithmic fairness considers the case of two binary groups, in the real world, candidates may have intersectional identities belonging to more than one protected group [26]. In the context of information retrieval, candidates might additionally be text or image information representing people [141], which could even have multiple people associated with candidate item [54]. Often, in practical cases the sensitive data may not be available for analysis, and evaluation strategies may depend on estimates of the likelihood that a candidate belongs to a certain group [136]. Or the problem setting may be more expansive, including for instance attributes such as the political leaning of news sources [133]. Questions around group identity are extremely important, unfortuanately they are beyond the scope of this dissertation. We discuss potential extensions of

our methodologies to multiple overlapping groups where appropriate. For simplicity, we henceforth consider two distinct groups in our fairness analysis.

7

Fair Ranking using Pairwise Error Metrics

This chapter presents fairness metrics for evaluating several notions of fairness for ranking. An auditing mechanism then is designed to produce nuanced diagnostics using our proposed fairness metrics. The efficacy of this approach is demonstrated using both a controlled case study and real-world scenarios, exposing trade-offs among fairness criteria and providing guidance in the selection of fair-ranking algorithms. This work was published in the paper [106].

> Caitlin Kuhlman, MaryAnn VanValkenburg, Elke Rundensteiner. FARE: Diagnostics for Fair Ranking using Pairwise Error Metrics. The Web Conference 2019

7.1 Proposed Fairness Metrics

To design a general approach for evaluating group error in rankings, we consider foundational approaches for comparing rankings [52]. One classic method is to sum the absolute difference in rank position between the true and predicted rankings for each object in the dataset (i.e., to use the Spearman footrule distance [53]). Another popular methodology uses the pairwise error, or Kendall Tau distance [94], by counting the number of inverted or discordant pairs of objects in the predicted ranking compared to the true ranking. These two classic approaches to measuring rank similarity have been shown to be *equivalent*, meaning the Kendall Tau is always within a constant factor of the Spearman footrule distance [53, 107]. Given this insight, the two metrics have been used interchangeably for tasks such as rank aggregation [60, 107].

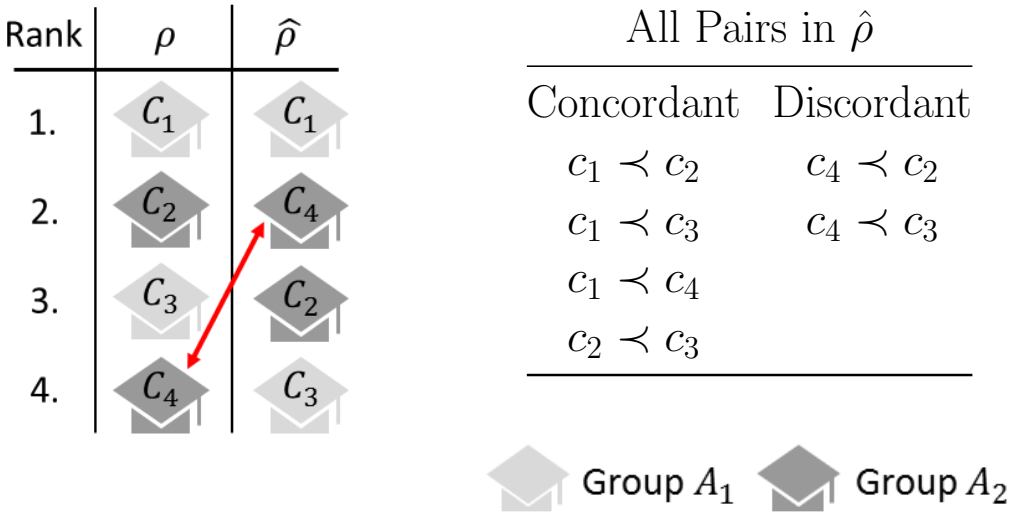


Figure 7.1: On the left is a true ranking of colleges ρ and predicted ranking $\hat{\rho}$ over two groups of colleges. The resulting discordant and concordant pairs are shown on the right.

For fairness assessment, the same reasoning applies in that either metric could be adapted to this task. However, since we are concerned with the comparative ranking outcomes for different groups, the pairwise approach provides a natural formulation.

Figure 7.1 shows the sets of concordant and discordant pairs between two rankings of colleges. We observe that any such ranking containing objects from two different groups, G_i and G_j, can be divided into three subsets of pairs: those containing only objects from group G_i, those containing only objects from G_j, and the set of "mixed" pairs containing one object from each group. These pairs can be *concordant*, meaning they are ordered in the same way in each ranking, or *discordant*, meaning their order is inverted in the predicted ranking. For our error metrics we will use pairwise comparisons to evaluate fairness by considering pairs where one groups is favored over another. We denote the total number of unordered pairs in a ranking over items in X as $\Phi(X) = |X|(|X|-1)/2$. Other sets of pairs are indicated with additional notation given in Table 7.1. $\mathrm{I}(\cdot)$ is the indicator function which equals 1 if $\cdot$ is true and 0 otherwise.

7.1.1 Rank Equality

The Equalized Odds criteria for classification measures fairness by the rate at which groups are falsely assigned to the preferred or non-preferred classes. When evaluating a ranking, there are no binary assignments by which to gauge preference. However, position in a ranking does indicate a preferred or undesirable outcome - the top of the ranking

Symbol	Value	Description
$\Phi(X)$	$\|X\|(\|X\|-1)/2$	Number of pairs in a ranking.
$\Phi_{i,j}(X)$	$\|G_i\|\|G_j\|$	Number of mixed pairs containing one item from each group G_i, G_j.
$\Phi_{i\prec j}(X)$	$\sum_{x_i\in G_i, x_j\in G_j} \mathrm{I}(\hat{\rho}(x_j) \prec \hat{\rho}(x_i))$	Number of mixed pairs favoring objects from group G_i over objects from G_j in the predicted ranking $\hat{\rho}$.
$\Phi^D_{i\prec j}(X)$	$\sum_{x_i\in G_i, x_j\in G_j} \mathrm{I}(\rho(x_j) \prec \rho(x_i) \text{ and } \hat{\rho}(x_i) \prec \hat{\rho}(x_j))$	Number of *concordant* pairs favoring G_i over G_j in $\hat{\rho}$.
$\Phi^C_{i\prec j}(X)$	$\sum_{x_i\in G_i, x_j\in G_j} \mathrm{I}(\rho(x_i) \prec \rho(x_j) \text{ and } \hat{\rho}(x_i) \prec \hat{\rho}(x_j))$	Number of *discordant* pairs favoring G_i over G_j in $\hat{\rho}$

Table 7.1: Notion denoting the number of pairs in different subsets of rankings.

being analogous to the positive class. When an object is overestimated by the model it is incorrectly assigned a more preferred position than in the true ranking. This is similar in effect to a false positive error made by a classifier. Accordingly, underestimating the position of an object in the ranking incorrectly penalizes it, as in a false negative. Following this principle, we compute the Rank Equality error for group G_i in terms of the number of discordant pairs which erroneously favor G_i over items from another group G_j in the predicted ranking. Our proposed metric in Definition 7.1 captures the rate at which objects from group G_i are incorrectly *overestimated* compared to objects from G_j. The Rank Equality error is normalized by the total number of mixed pairs ensuring that the error falls in a range of $[0, 1]$. Normalization creates an interpretable measure of preference and accounts for any imbalance in the size of the groups.

Definition 7.1. ***Rank Equality Error.*** *Given a ground truth ranking ρ and a predicted ranking $\hat{rho}$ of items $x_i \in X$ belonging to two mutually exclusive groups G_1 and G_2, where $\phi^D_{i\prec j}(X)$ denotes the number of discordant pairs which favor the target group G_1 over G_2 in $\hat{\rho}$, $i \neq j$, Rank Equality for G_i is computed as:*

$$Req_{G_1}(\rho, \hat{\rho}) = \frac{\Phi^D_{i\prec j}(X)}{\Phi_{i,j}(X)}$$

Rank Equality dictates that no group should be unfairly privileged or penalized compared to another group. As an example, consider the rankings shown in Figure 7.1. To compute the Rank Equality errors for groups G_1 and G_2, we count the number of discordant pairs where an item from one group is favored over the other. Four pairs contain an object from each group: $(c_1, c_2), (c_1, c_4), (c_2, c_3), (c_3, c_4)$. One of these pairs (c_3, c_4) is

discordant and favors G_2, since $\hat{\rho}(c_4) \prec \hat{\rho}(c_3)$ and $\rho(c_3) \prec \rho(c_4)$. Thus $Req_{G_2}(\rho, \hat{\rho}) = \frac{1}{4}$. No discordant pairs favor G_1, so $Req_{G_1} = 0$.

7.1.2 Rank Calibration

Calibration is used to evaluate probabilistic classifiers in terms of the confidence of the model, using the mean squared error between predicted likelihood of assignment in the positive class and an estimated "true" probability [31]. Applied as an FC, calibration checks how well the classifier predicts objects in each group. To evaluate the calibration of a ranking $\hat{\rho}$ for a group G_i, we propose to measure error in predicted rank position by counting the number of discordant pairs which contain at least one member of G_i, as given in Definition 7.2. This captures the overall error made for items in the group. The value is normalized by the total number of pairs containing objects from G_i.

Definition 7.2. ***Rank Calibration Error.*** *Given a ground truth ranking ρ and a predicted ranking $\hat{rho}$ of items $x_i \in X$ belonging to two mutually exclusive groups G_i and G_j, where $\Phi_i^D(X)$ denotes the number of discordant pairs containing at least one object from the target group G_i, Rank Calibration for G_i is computed as:*

$$Rcal_{G_i}(\rho, \hat{\rho}) = \frac{\Phi_i^D(X)}{\Phi(X) - \Phi(G_j)}$$

In our example, the pairs containing objects from group G_2 in Figure 7.1 are (c_1, c_2), $(c_1, c_4), (c_2, c_3), (c_2, c_4), (c_3, c_4)$. Pairs (c_2, c_4) and (c_3, c_4) are both discordant, therefore following Definition 7.2, the rank calibration error is $Rcal_{G_2}(\rho, \hat{\rho}) = \frac{2}{5}$. Five pairs contain items from G_1: $(c_1, c_2), (c_1, c_3), (c_1, c_4), (c_2, c_3), (c_4, c_3)$, but only (c_4, c_3) is discordant, so $Rcal_{G_1} = \frac{1}{5}$

7.1.3 Rank Parity

Finally, we apply pair inversion to design a metric which falls into the statistical parity class of fairness criteria, like those explored in previous work on fair ranking [34, 155, 161]. Here, the goal is to ensure fair representation of members of each group among objects given a favorable rank position. We propose to capture this idea by counting the pairs in which one group is favored over the other in the learned ranking, regardless of their positions in the true ranking. We again normalize by the total number of mixed pairs in the learned ranking.

Definition 7.3. ***Rank Parity Error*** *Given a predicted ranking $\hat{\rho}$ of items $x_i \in X$ belonging to two mutually exclusive groups G_i and G_j, Where $\Phi_{i \prec j}(X)$ is the number of mixed pairs of objects which favor the target group G_i over G_j in $\hat{\rho}$, $i \neq j$, Rank Parity for G_i is computed as:*

$$Rpar_{G_i}(\rho, \hat{\rho}) = \frac{\Phi_{i \prec j}(X)}{\Phi_{i,j}(X)}$$

In Figure 7.1, two pairs in $\hat{\rho}$ (c_1, c_4) and $(c_1, c2)$ favor group G_1 over G_2 and two pairs (c_2, c_3) and (c_4, c_3) favor G_2 over G_1 out of the four possible mixed pairs. Therefore, $Rpar_{G_1}(\rho, \hat{\rho}) = Rpar_{G_2}(\rho, \hat{\rho}) = \frac{1}{2}$. This matches our intuition of parity, since the groups are still somewhat evenly distributed through the ranking $\hat{\rho}$ in spite of the incorrect placement of c_4.

7.2 Relationships between Fair Error Metrics

We now analyze our metrics to understand their interrelationships and scope of applicability. Given a ranking $\hat{\rho}$ over g groups, there are g^2 ways of choosing two objects from the ranking, allowing for group repetition. These pairs may be either concordant or discordant, resulting in $2g^2$ types of pairs. Table 7.2 shows the categories of pairs that can be formed for $g = 2$ groups. The colors in the table correspond to the colors in the Venn diagram in Figure 7.2, which illustrates the relationship between the types of pairs used to compute our proposed error metrics for a single group, G_1.

Since our fairness analysis is concerned with the relative error made for each group, within-group concordant pairs are not considered when computing error metrics. All other types of pairs are included in the definition of at least one error metric. Discordant mixed pairs are used to compute all three error metrics. These pairs of objects intuitively capture the *major disparity between groups*: cases where one group is erroneously favored over the other. We define this as Rank Equality. Rank Calibration instead measures the *total error* for each group. This metric counts all pairs containing objects from a

Discordant	Concordant
$G_1 \succ {G_1}^D$	$G_1 \succ {G_1}^C$
$G_2 \succ {G_2}^D$	$G_2 \succ {G_2}^C$
$G_1 \succ {G_2}^D$	$G_1 \succ {G_2}^C$
$G_2 \succ {G_1}^D$	$G_2 \succ {G_1}^C$

Table 7.2: Categorization of pairs in $\hat{\rho}$ over two groups G_1, G_2.

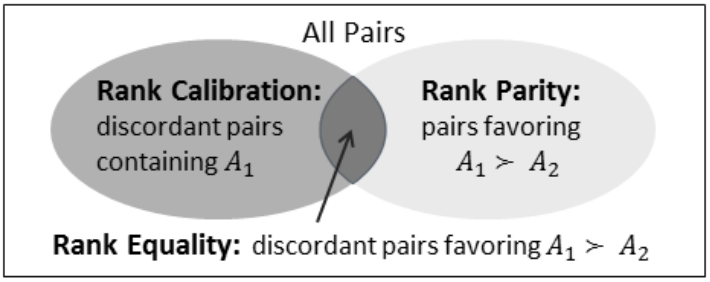


Figure 7.2: Relationship between the types of pairs used to compute the error for group G_1 (corresponding to Table 7.2).

single group, capturing within-group as well as across-group errors. Finally, Rank Parity considers the *total advantage* of one group over the other.

FC are compatible in extreme cases. In a perfect prediction there is no error between $\hat{\rho}$ and ρ. In this case $Req = Rcal = 0$ for all groups, since no pairs are discordant. The corresponding FC deem $\hat{\rho}$ is fair since the group errors are identical. The *Rpar* in this case will simply measure the relative advantage of each group in the true ranking. It may be considered fair or unfair depending on the distribution of the objects in each group. However, this is independent of the other metrics, and therefore, it is possible for a perfect prediction to satisfy all three criteria. In the worst case, $\hat{\rho}$ ranks the objects in the reverse order from ρ. In this case $Rcal = 1$ since all pairs are discordant. Each group is predicted with the same amount of error, therefore by the Rank Calibration FC, $\hat{\rho}$ is considered fair. In this case no pairs are concordant, therefore $Req = Rpar$. Whether the ranking is considered fair according to the corresponding FC again depends on the distribution of groups throughout the ranking.

7.3 FARE: Fair Auditing based on Rank Error

Using a single fairness score to describe a ranking provides a coarse assessment of fairness. To understand the entire ranking of items from the preferred positions at top of the ranking to the lowest ranked objects, we next design a non-parametric strategy to assess rankings using our proposed pairwise fairness metrics. Our FARE framework (for Fair Auditing based on Rank Error) generates *sequences of within-range errors* for each group. The differences in these sequences tell a richer story than would a single value for each group, revealing disparity throughout the entire ranking.

7.3.1 Methodology

To start, FARE sorts the data according to the predicted ordering $\hat{\rho}$ and bins it into k subsets $\langle B_1, B_2, \ldots B_k \rangle$. Error metrics are then applied to the objects in each bin. In the case of two groups A_1 and A_2 we evaluate errors $l_{1i} = L_{A_1}(\beta_i, \hat{\beta}_i)$ and $l_{2i} = L_{A_2}(\beta_i, \hat{\beta}_i)$ for the data in each bin B_i. This produces two error sequences: $S_1 = \langle l_{11}, l_{12}, \ldots l_{1k} \rangle$ and $S_2 = \langle l_{21}, l_{22}, \ldots l_{2k} \rangle$. An equi-width binning strategy compares the top-k ranked items across both groups in the first bin, the next k in the next bin, and so on. An equi-depth strategy is also possible, where each bin measures how well the ranking predicts $\frac{|A_i|}{k}\%$ of items from each group.

If the number of bins k is so large that there are only a few objects in each, then the sequence of error measurements may exhibit a high degree of variance. This could exaggerate differences between groups in the case where one is a minority. On the other hand, if bins contain many objects, the result is a coarse estimate. To capture the error at a sufficient number of positions throughout the ranking while achieving a reasonable bin size, we adopt a *sliding window approach*. This introduces a smoothing transformation over the data to account for high variance across bins. Each consecutive bin of size w overlaps the previous, offset by a fixed step size $s < w$. The first bin B_1 contains objects $\{x_1, x_2, .., x_w\}$, ordered according to their predicted positions $\hat{\rho}(x_i)$, the second bin $B_2 = \{x_{s+1}, x_{s+2}, \ldots x_{s+w}\}$, and so on. Each error sequence S_i contains $\lfloor \frac{|A_i|}{s} \rfloor$ bins.

7.3.2 Diagnostics for Analyzing Fairness

The next step in the auditing procedure is to compare the error sequences S_1 and S_2 produced by our FARE framework to see if they are similar, and therefore meet the fairness criterion, or if they differ in ways which indicate an unfair ranking. To facilitate this, FARE offers *audit plots*. Similar to reliability diagrams for assessing the calibration of classifiers [31], these visual depictions reveal differences in the shapes, patterns and values of the error sequences. Since our metrics are normalized, the y-axis of each plot has a fixed range of $[0, 1]$, providing an easily interpretable snapshot of the error sequences generated during the audit process.

Audit plots are augmented by compact statistics, or *fairness scores*, indicating whether the ranking model satisfies the FC. Conceptually, any diagnostic metrics comparing the sequences can be plugged into the framework. We employ a distance diagnostic $dist(S_1, S_2) = \frac{1}{k} \sum_{i=1}^{k} |l_{1i} - l_{2i}|$ to summarize the similarity of the error sequences as a single value. These scores can then be thresholded to flag unfair cases where the average

magnitude of error for one group is much larger than the other.

7.3.3 Complexity

A simple pair counting algorithm can be used to compute each of our proposed error metrics in $O(nlog(n))$ time using an adaptation of the mergesort algorithm. Performing an audit using the FARE methodology can therefore also be done in logarithmic time, requiring $O(n/s(wlog(w))$ time for step size s and window size w to compute the error sequences for each group. In cases where performance is an issue, we can improve the pair counting procedure to run in $O(n\sqrt{log(n)})$ time [36].

7.4 Evaluation

We illustrate the power of our FARE framework to uncover different types of systematic disparity between groups. In our first study, we apply FARE to distinct cases of unfairness. Each FC is considered in turn to identify different manifestations of unfairness. In a second study, we use FARE to audit post-processing techniques from the literature designed to correct existing rankings.

7.4.1 Auditing Diverse Unfair Scenarios

Dataset. We generate a random dataset X ordered by a utility score between 0 and 1 for each object to produce our "true" ranking ρ. A randomly assigned binary protected attribute divides the data into two groups G_1 and G_2 of roughly even size. We generate a "baseline" ranking $\hat{\rho}$ by adding a small amount of Gaussian noise to simulate irreducible error in a predictive model. We then design a family of rankings by adding additional degrees of error $p \leq 1$ for one or both groups. Unfair *underestimation* is simulated by scaling the utility score for each $x \in G_1$ by a random factor between p and 1. For *overestimation*, G_2 is scaled by a random factor between 1 and $1 + p$. In the third type of ranking, G_1 is underestimated and G_2 is overestimated.

Figure 7.3 shows the audit plots for these different scenarios for data of size $|X| = 10000$ and parameters $w = 350, s = 10$. Trend diagnostics are given in Table 7.4 on the right and distances in Table 7.5. For the baseline case in plot (a), the trends are close to zero, indicating error is consistent throughout the ranking, and the distances between sequences are small. Plots (b) (c) depict the result of underestimating group

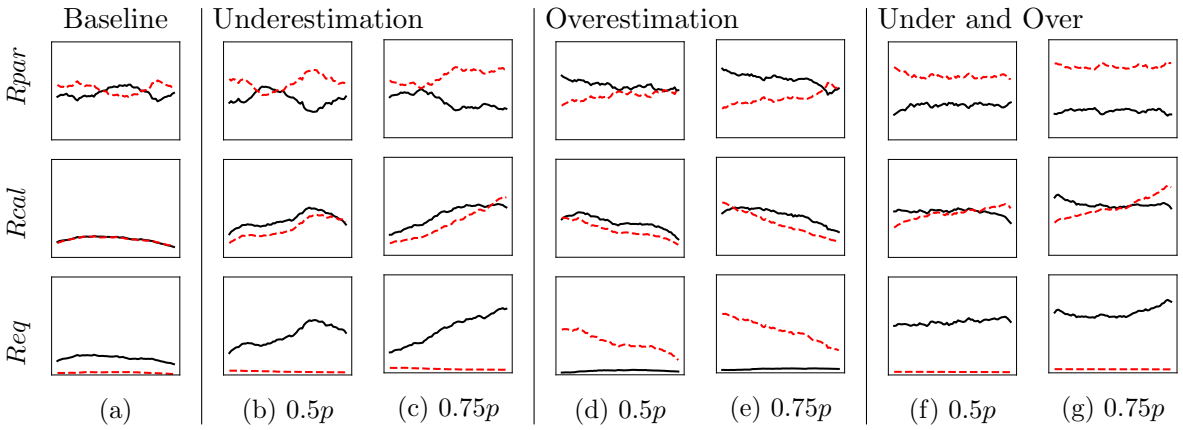


Figure 7.3: Audit plots for case study on varying degrees of error illustrate how errors (plotted on the y-axis and normalized between 0 and 1), manifest throughout the ranking. The x-axis represents the sliding window moving from highly ranked items on the left to the lowest on the right. p indicates the amount of error introduced. Errors for group G_1 are shown as a solid black line, group G_2 as dashed red line. Top row: Rank Parity, middle row: Rank Calibration, bottom row: Rank Equality.

Rpar	—	0.01	-0.10	-0.21	-0.12	-0.19	0.04	-0.01
	--	-0.01	0.10	0.21	0.12	0.19	-0.04	0.01
Rcal	—	-0.06	0.18	0.34	-0.20	-0.23	-0.05	-0.06
	--	-0.06	0.29	0.50	-0.25	-0.40	0.19	0.34
Req	—	-0.06	0.31	0.50	0.01	0.01	0.08	0.11
	--	-0.01	-0.02	-0.02	-0.28	-0.37	-0.00	-0.00
		a	b	c	d	e	f	g

Figure 7.4: Trend diagnostics for case study scenarios shown in Figure 7.3.

Rpar	0.06	0.19	0.25	0.11	0.21	0.30	0.45
Rcal	0.00	0.08	0.10	0.08	0.10	0.06	0.09
Req	0.14	0.38	0.42	0.29	0.40	0.52	0.60
	a	b	c	d	e	f	g

Figure 7.5: Distance diagnostics for case study scenarios shown in Figure 7.3.

G_1 by factors of $p = 0.5$ and 0.75, respectively. Plots (d) (e) show overestimation of G_2 while (f) (g) show when G_1 is underestimated and G_2 overestimated.

Rank Parity. Along the top row in Figure 7.3, we see *Rpar* sequences for each group. As error is introduced for rankings (b) - (g), distance between the sequences grows. From the audit plots we observe that when one group is *underestimated*, the sequences

deviate. When one group is *overestimated*, the sequences start far from each other and then converge. This illustrates a way in which error manifests differently throughout the ranking. This confirms that it is important to consider the entire ranking, not just the prefix. In the final case, where both groups are mis-estimated, we observe overall trends which are flat. While this could be misinterpreted as a case with no disparity, the larger distance scores in Table 7.5 clearly indicate the higher degree of unfairness.

Rank Calibration. Audit plots and trend diagnostics for the *Rcal* error sequences, shown in the middle row above, reveal that when a single group is *underestimated*, the Rank Calibration error sequences for both groups trend up, and when a single group is *underestimated*, the trends slope downward. When both groups are mis-estimated the trends are flat. Table 7.5 shows that, in all cases, the *Rcal* error sequences for each group retain a similar distance. In this last case, diagnostic scores are not sufficient to convey unfairness without audit plots. We observe that the magnitude of error is informative, since the baseline is around 0.1 while the mean errors for $p = 0.5$ and $p = 0.75$ are close to 0.5.

Rank Equality. The bottom row of Figure 7.3 shows the error sequences for the *Req* error metric. As more error is introduced, the sequences for each group diverge. The baseline "fair" rankings have scores of 0.14, while the distances increase up to 0.60 when both groups are mis-estimated with $p = 0.75$. We also observe that when error is introduced for only one group, the slope of the corresponding error sequence again trends up for *underestimation* and down for *overestimation*, which is captured by the trend scores in Table 7.4.

Overall, Rank Calibration is the least sensitive metric to the types of unfairness presented. Applying a fairness threshold of 0.11 to the distance scores in Table 7.5 will flag all cases where the groups are treated unfairly using the Rank Parity FC and Rank Equality FC. However, none of the cases are identified by the Rank Calibration FC, which requires a lower threshold of 0.06.

7.4.2 Auditing Rank Correction Methods

Next we apply FARE to rankings generated using recently proposed rank correction techniques. We reproduce a subset of experiments presented by Zehlike et al. [161] using implementation and data provided by the authors.

Fair Correction Methods. "FA*IR" rankings are generated using the algorithm proposed in [161] to create a fair top-k prefix ranking. The rankings target a user-specified minimum proportion of the minority group, subject to a statistical significance test. The proportion is indicated in the method name, e.g. FA*IR2 for 20%. Here we use the same proportions as the authors, chosen to be close to the actual group ratio over the entire dataset. The "Feldman" method was proposed by Feldman et al. [65] as a pre-processing step for fair classification in which data are ranked. In this method the utility scores for objects in the minority group are adjusted to match the distribution of the majority. We compare these rankings against a baseline of the true ranking simply compared to itself.

Datasets. The Statlog German Credit Dataset [84] is utilized, with a "true" ranking of people created according to credit-worthiness. Three "fair" rankings are then created using $age < 25$, $age < 35$ and $gender$ = female as protected group attributes. Prefix rankings with $k = 100$ are generated. (Audit parameters for this dataset are $w = 30, s = 10$). The COMPAS recidivism dataset published by ProPublica in their investigation of racial bias in the criminal justice system is also utilized [8]. The dataset is ranked according to the COMPAS scores indicating the likelihood of re-offending for the "true" ranking with $k = 1000$. "Fair" rankings are generated according to groups $race$ = African American and $gender$ = male. (Audit parameters $w = 100, s = 10$).

Metrics. We produce audit plots using our proposed metrics $Req, Rcal$ and $Rpar$, and summarize the results using FARE distance diagnostics. In their experiments, Zehlike et al. [161] use a number of metrics to gauge the tradeoff between parity and prediction accuracy. We include two metrics for comparison: $NDCG$: normalized discounted cumulative gain [90] (commonly used in search), and *rank drop*: the maximum number of positions lost by an object. Table 7.3 summarizes the FARE diagnostics for our experiments. The rankings deemed most fair in this audit are highlighted in bold. Asterisks mark the conclusions which align with the analysis in [161]. For three out of five rankings FA*IR outperforms Feldman, satisfying multiple fairness concerns.

7.3 summarizes the FARE diagnostics for our experiments. The rankings deemed most fair in this audit are highlighted in bold. Asterisks mark the conclusions which align with the analysis in [161]. For three out of five rankings FA*IR outperforms Feldman, satisfying multiple fairness concerns. Discussion. The impact of both the FA*IR and Feldman rank correction techniques on statistical parity concerns is apparent, as measured by our Ranking Parity FC. For instance, for the German Credit dataset using

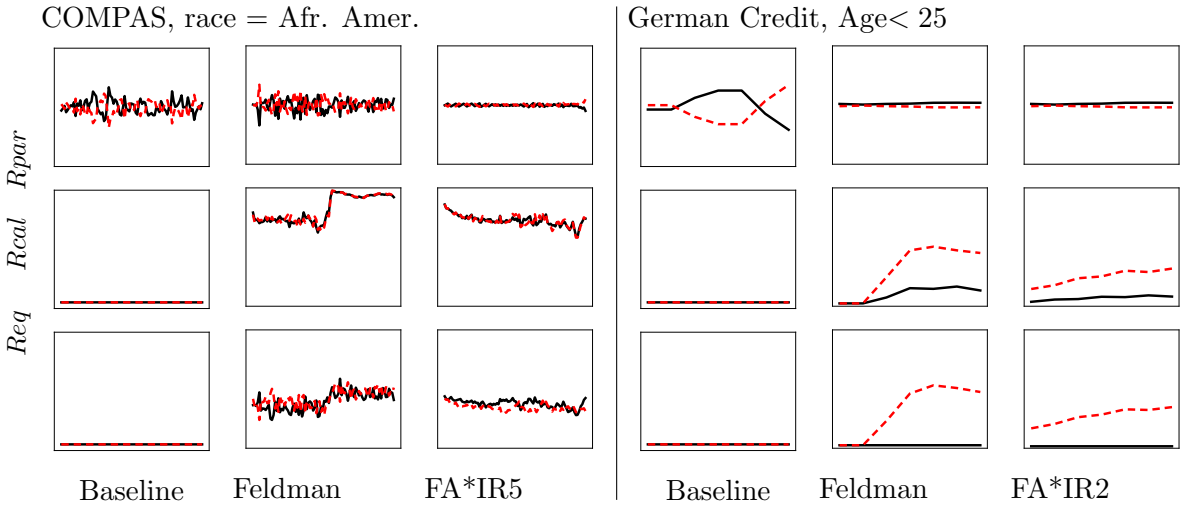


Figure 7.6: Audit plots for rank correction methods. Errors are plotted on the y-axis and normalized between 0 and 1. The x-axis represents the sliding window moving from highly ranked items on the left to the lowest on the right. Errors for the group defined by the sensitive attribute are depicted the dashed red line. Top row: Rank Parity, middle row: Rank Calibration, bottom row: Rank Equality.

			FARE				
Dataset	**Group**	**Method**	*Rpar*	*Rcal*	*Req*	**NDCG**	**Drop**
German Credit k=100	age< 25	Baseline	0.18	0.00	0.00	1.00	0
		Feldman	**0.03**	0.22	0.31	1.00	8
		FA*IR2	**0.03***	**0.18**	**0.26**	1.00	7
German Credit k=100	age< 35	Baseline	0.21	0.00	0.00	1.00	0
		Feldman	**0.04**	**0.05**	**0.22**	0.99	36
		FA*IR6	0.11	**0.05**	0.43	0.99	30
German Credit k=100	gen=f	Baseline	0.30	0.00	0.00	1.00	0
		Feldman	**0.03**	**0.11**	**0.28**	1.00	8
		FA*IR7	0.1	0.15	0.33	1.00	0
COMPAS k=1000	race	Baseline	0.09	0.00	0.00	1.00	0
		Feldman	0.08	**0.01**	0.08	0.98	393
		FA*IR5	**0.02***	**0.01**	**0.04**	0.99	319
COMPAS k=1000	gen=m	Baseline	0.13	0.00	0.00	1.00	0
		Feldman	0.09	0.03	0.09	1.00	294
		FA*IR8	**0.02***	**0.01**	**0.03**	1.00	161

Table 7.3: Fairness evaluation for rank correction methods. FARE distance diagnostics are shown in the center, and compared to standard error metrics.

$age < 25$, *Rpar* distance is 0.18 in the baseline "true" ranking. Both methods are able to reduce this to 0.03. The degree to which error is introduced as a result of the correction

algorithm is reflected in the *Req* and *Rpar* scores. By comparison, the NDCG metric is not sensitive to the rank correction methods, and therefore not expressive enough to capture unfairness. The rank drop values tend to align with the FARE diagnostics. However, this value is not very interpretable. We cannot observe which group had the farthest drop, or whether the position of many items dropped.

For such nuanced analysis we turn to our FARE audit plots, shown in Figure 7.6. For the German Credit dataset, we can visually discern that both FA*IR and Feldman introduce similar error patterns. The *Req* and *Rcal* errors increase throughout the ranking for both groups. For the COMPAS dataset, we observe that the patterns and magnitude of error throughout the rankings are similar for both groups. By our FC, this implies the correction methods are introducing error in a fair manner. Feldman shows a jump midway through the *Rcal* sequence while the FA*IR error decreases through the ranking. FARE compliments the use of these ranking methods by providing this in-depth view of the treatment of each group.

8

Comparative Study of Statistical Parity Metrics for Ranking

To address the need for a nuanced understanding of evaluation metrics for fair ranking, in this Chapter we study one class of metrics in-depth – **statistical parity metrics**. These metrics provide a basic foundation for understanding unfair group advantage in rankings, providing the basis for much fair ranking metric design. We perform a survey of proposed statistical parity metrics categorized by strategies for measuring group advantage. Then we propose a conceptual framework for metric comparison and demonstrate metric behavior under various assumptions about the relative advantage of the groups. This work being prepared as the following submission to a major conference:

> Caitlin Kuhlman, Walter Gerych, Elke A. Rundensteiner, *Measuring Group Advantage: A Comparative Study of Fair Ranking Metrics.*

8.1 Survey of Statistical Parity Metrics

Statistical parity is a simple approach to group fairness which requires that different groups receive fair proportions of favorable outcomes. This could mean that each group receives an equal share, a minimum according to some external target, or a proportion based on the size of the groups in the overall population or dataset. Often one group is considered to be at a disadvantage which requires some affirmative intervention to ensure fairness. We refer to such a "protected group" throughout our survey, denoted G_p. Definition 8.1 gives a typical definition of statistical parity [39], which was first proposed as a fairness criterion for classification tasks [58].

Definition 8.1. ***Statistical Parity for Classification:*** *Given a dataset X of candidates belonging to mutually exclusive groups $G_i \in G$ and a binary classifier $f(x) = \hat{y}$ which assigns each item $x \in X$ to a class in $\{0, 1\}$, where $\hat{y} = 1$ denotes the preferred outcome for item x, the predictor satisfies statistical parity if the following condition is met for all groups $G_i, G_j \in X, i \neq j$.*

$$P(\hat{y} = 1 \mid x \in G_i) = P(\hat{y} = 1 \mid x \in G_j)$$

Statistical Parity in Rankings. The metrics we consider in this work aim to establish a similar statistical parity requirement for rankings. However, in this setting there is no binary class assignment with which to evaluate the outcomes for the groups. Clearly, being ranked toward the top is a better outcome than being ranked near the bottom, but this determination is inherently relative. Therefore proposed metrics for statistical parity in rankings draw on traditional rank evaluation methods for measuring the *relative advantage* of each group being ranked. Next we review and categorize proposed metrics according to the different approaches for measuring group advantage.

8.1.1 Top-k Measures

A popular method for identifying a favorable outcome in a ranking is by inclusion in a top-k prefix of the ranking. This approach is intuitive and interpretable, since for many tasks the top k rank positions directly correspond to a good outcome for the candidates being ranked, e.g. the top 5 job applicants are invited to interview for a position, or the top 10 documents appear on the first page of search results. Therefore a number of recent works on fair ranking require fair representation of groups in the top-k rank positions [10, 34, 70, 155, 161]. Definition 8.2 gives a probabilistic formulation of top-k statistical parity for rankings.

Definition 8.2. ***Top-k Parity:*** *Given a ranking $\rho = [x_1 \prec x_2 \prec ... \prec x_n]$ of candidates belonging to mutually exclusive groups $G_i \in X$, and $0 \leq k \leq n$, the ranking satisfies top-k parity if the following condition is met for all groups $G_i, G_j \in X, i \neq j$.*

$$P(\rho(x) \leq k \mid x \in G_i) = P(\rho(x) \leq k \mid x \in G_j)$$

A variety of top-k based formulations of statistical parity have been proposed, however not all define a numeric measure of fairness. We evaluate summary statistics which measure whether a ranking adheres to the statistical parity goal [70, 155]. Since top-k

metrics are highly dependent on the choice of k, a cumulative strategy for multiple k is typically used, modeled on the popular nDCG metric [90]. Metric scores are weighted by some discounting function $v(k)$ and aggregated. The goal is to give more emphasis to fair or unfair outcomes at the top of the ranking. The aggregated scores may also be normalized to lie between $[0, 1]$ by computing the ideal (maximum) value Z. In our analysis for simplicity we use a logarithmic discounting function over fixed intervals of top-k prefixes proposed by Yang and Stoyanovich [155] for all top-k metrics[1], such that the final fairness score for a given metric M is computed as:

$$\frac{1}{Z} \sum_{k=10,20,10...}^{n} \frac{1}{log_2(k)} M \tag{8.1}$$

Proposed top-k metrics M include:

- **Normalized discounted difference (rND) [155]** is evaluated only for the protected group G_p. Fairness is measured as the difference between the representation of G_p in the top-k and in the entire ranking.

$$rND(\rho) = P(x \in G_p \mid \rho(x) \leq k) - P(x \in G_p) \tag{8.2}$$

- **Normalized discounted ratio (rRD) [155]** compares ratios of outcomes for the protected group G_p and a non-protected group G_i for the top-k and the entire ranking.

$$rRD(\rho) = \frac{P(x \in G_p \mid \rho(x) \leq k)}{P(x \in G_i \mid \rho(x) \leq k)} - \frac{P(x \in G_p)}{P(x \in G_i)} \tag{8.3}$$

- **Skew@k [70]** computes the logarithmic ratio of outcomes in the top-k versus the entire list. This metric is also evaluated for a single group G_i, however alternative versions *minskew* and *maxskew* are proposed for whichever group has the maximum or minimum skew value.

$$skew_{G_i}@k(\rho) = log(\frac{P(x \in G_i \mid x \leq k)}{P(x \in G_i)}) \tag{8.4}$$

- **Kullback-Leibler divergence (rKL)** was first proposed for evaluating statistical parity by Yang and Stoyanovich for the case of two groups [155] and then extended

[1]We note that in the original paper by Geyik et al. [70], skew metrics are computed only on a single fixed top-k, and $NDKL$ is aggregated over all k.

to the more general case of multiple groups by Geyik *et al.*. [70]. In the general case, rKL metric is computed as:

$$rKL(\rho) = KL(P||Q) \tag{8.5}$$

where $P = P(\rho(x) \leq k \mid x \in G_i)\ \forall G_i$, the proportion of each group in the top-k items, and $Q = P(x \in G_i)\ \forall G_i$, the proportion of each group in the entire ranking.

8.1.2 Exposure

Singh and Joachims [141] proposed a statistical parity metric in the context of an IR-focused framework. In this setting, they consider the favorable outcome to be "exposure", a measure of the attention given to a candidate at a particular rank position. In this general formulation the attention score could be given by a discounting function $v(k)$ or some other measure of the importance of each position, for instance learned from implicit user feedback. Definition 8.6 gives a measure of advantage for group G_i based on the average importance of the rank positions assigned to each candidate $x \in G_i$.

$$exp_{G_i}(\rho) = \frac{1}{|G_i|} \sum_{x \in G_i} v(\rho(x)) \tag{8.6}$$

The overall fairness of a ranking with two groups G_i and G_j is determined as the absolute difference in the exposure scores.

$$exp(\rho) = \left| exp_{G_i}(\rho) - exp_{G_j}(\rho) \right| \tag{8.7}$$

Rankings where each group receives equal exposure are deemed fair. In our evaluation we consider two versions of exposure metrics. $expDCG$ uses a logarithmic discount to model attention such that $v(k) = \frac{1}{log(k+1)}$ for each position k. We also consider $expRR$ which uses a reciprocal rank function where $v(k) = \frac{1}{k}$.

8.1.3 Pairwise Measures

Finally, we consider metrics which use the pairwise advantage of each group to evaluate parity [17, 106, 121]. These metrics are modelled on the classic Kendall Tau distance between rankings [94] which counts pair inversions between two lists. Definition 9.4 gives a pairwise formulation of statistical parity.

Definition 8.3. ***Pairwise Statistical Parity:*** *Given a ranking* $\rho = [x_1 \prec x_2 \prec ... \prec x_n]$ *of candidates belonging to mutually exclusive groups* $G_i \in X$*,* ρ *satisfies pairwise statistical parity if the following condition is met for all groups* $G_i, G_j \in X, i \neq j$

$$P(x_i \prec_\rho x_j \mid x_i \in G_i, x_j \notin G_i) = P(x_i \prec_\rho x_j \mid x_i \in G_j, x_j \notin G_j)$$

Pairwise metrics compute the advantage for a single group based on the number of pairwise comparisons it wins against items from other groups in the ranking. We consider the *Rpar* metric introduced in Section 7.1.3 in our analysis. Equation 8.8 restates this pairwise statistical parity requirement. The metric is normalized by the number of pairs in the ranking containing candidates from different groups. Here $I(\cdot)$ is the indicator function which evaluates to 1 if $\cdot$ is true and 0 otherwise.

$$Rpar_{G_i}(\rho) = \frac{1}{|G_i||G_j|} \sum_{x_i \in G_i} \sum_{x_j \in G_j} I(\rho(x_i) \prec \rho(x_j)) \tag{8.8}$$

The overall fairness of a ranking with two groups G_i and G_j is determined as the absolute difference in the *Rpar* scores.

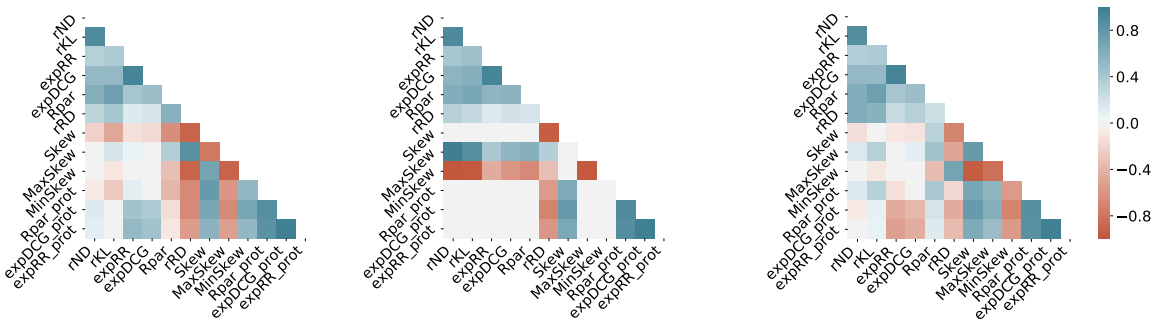
$$Rpar(\rho) = \left| Rpar_{G_i}(\rho) - Rpar_{G_j}(\rho) \right| \tag{8.9}$$

8.1.4 Correlation Analysis

As an initial investigation into the relationships among these metrics we perform a correlation analysis. We generated 1000 random rankings with $n = 100$ candidates from two groups of varying size and compute the statistical parity metrics for each ranking. For the exposure ($expDCG, expRR$) and pairwise ($Rpar$) metrics we report both the absolute difference of the advantage for each group, as well as the metric computed only for the protected group G_p. For this analysis the top-k metrics are not normalized.

Figure 8.1 shows the Pearson correlation between each pair of metrics. Correlation values with significance less than $p = 0.05$ are set to 0. In the upper left hand corner of the heatmaps, The rND, rKL, rRD, $expDCG$, $expRR$, and $rPar$ are positively correlated with each other, across rankings with different sized groups. This core group of metrics appear to share similar behavior, perhaps capturing the same overall fairness, and are not sensitive to group size. Pairs of metrics with particularly strong positive correlation are $rKLR$ and rND, and $expDCG$ and $expRR$.

Interestingly, when the groups are unbalanced the rest of the metrics are somewhat



(a) Protected group 20%. **(b)** Protected group 50%. **(c)** Protected group 80%.

Figure 8.1: Pearson correlation between statistical parity metrics for ranking is shown for 1000 randomly generated rankings of 100 candidates belonging to two distinct groups. Correlation values with significance $p > 0.05$ are omitted.

correlated with the first group, but not when the groups are the same size. In this case only the rRD metric from the core metrics correlates with the with $skew$, $rPar_{G_p}$, $expDCG_{G_p}$, and $expRR_{G_p}$. On the other hand, $maxskew$ and $minskew$ now much more strongly correlate with the other core metrics.

This suggests that the skew metrics along with those computed only for the protected group tell half the story, and do not perform consistently, particularly when no group has a majority. They may be capturing a different type of fairness than the statistical parity definitions targeted. We therefore mainly focus on the core group of metrics identified in this analysis for the rest of the paper, and revisit the other metrics in our empirical evaluation in Section 7.4.

8.2 Framework for Fair Ranking Metric Comparison

Our correlation analysis suggests that a core set of state-of-the-art fair ranking metrics share similar behavior: rND, rKL, rRD, $expDCG$, $expRR$, and $Rpar$. Each metric compares the relative advantage of the groups using different strategies. However, we still don't know how this group advantage relates to unfairness in the rankings - hence many approaches seem equally compelling. We now propose a common framework for comparing the behavior of the metrics with respect to unfair group advantage. First, rather than focus on any discrete single ranking, we henceforth describe the behavior of the metrics in expectation over distributions of rankings.

8.2.1 Probabilistic Assignment Matrix

We propose the use of a matrix $R \in \mathbb{R}^{NxN}$ to represent distributions over rankings, where N is the number of candidates to be ranked. Each row of R represents a position in the ranking, and each column represents a candidate. Each entry $R_{i,j}$ gives the probability that candidate x_j is assigned rank position i, as expressed in Equation 8.10. As each element of R represents the probability that a given element is in a given position, the rows and columns must each add up to 1 and thus we call R doubly stochastic. In the case of a single discrete ranking, R is a binary matrix where $R_{i,j} = 1$ iff $\sigma(x_j) = i$, and $R_{i,j} = 0$ otherwise.

$$R_{i,j} = P(\sigma(x_j) = i) \tag{8.10}$$

8.2.2 Modeling Group Advantage

Fairness metrics are predicated on the belief that one ranking can give a more preferred outcome to one group than another. Therefore we propose to model this unfair advantage as a random variable α which can take on some range of values. For convenience we can choose $\alpha \in [0, 1]$ where a score of 0 means that a group is at a complete disadvantage, and 1 indicates a total advantage over other groups. As is common in group fairness analysis, we assume that the unfairness impacts all members within one group similarly, but that different groups have different levels of advantage. For simplicity in our analysis we consider a single value for α representing the advantage of the protected group G_p, with the understanding that this equivalently implies an advantage or disadvantage for other non-protected groups.

To model this in our framework we now impose some additional structure on R to represent the probability of candidates being assigned to each position as a function of advantage α. Let us assume that each entry in R is a function such that:

$$R_{i,j} = f_{i,j}(\alpha), \quad f : [0, 1] \to [0, 1] \tag{8.11}$$

This framing reflects the fact that group advantage does not impact an entire ranking uniformly – it varies if evaluated at each position in the ranking. For instance, at position $i = 1$, if G_p has a large advantage and many possible candidates to choose from, it is highly likely that a protected candidate will be assigned to the top spot. However, for a lower rank position most protected candidates will have been assigned to positions above, and a non-protected item will now be more likely to be assigned. Therefore although

the overall advantage does not change, $f_{i,j}(\alpha)$ at different rank positions i gives different likelihoods of assignment for x_j.

8.2.3 Expressing Statistical Parity Metrics in Terms of Group Advantage

Next we represent the core set of statistical parity metrics in our framework as functions of the advantage α.

Top-k metrics. Each of the top-k metrics measures some distance or divergence between the proportion of the protected group in the top-k, and the proportion of the protected group over the entire ranking. We compute these values in terms of the doubly stochastic matrix R as follows:

$$P = P(x \in G_p \mid \rho(x) \leq k) = \frac{1}{k} \sum_{i=1}^{k} \sum_{j \in G_p} f_{i,j}(\alpha) \tag{8.12}$$

$$Q = P(x \in G_p) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j \in G_p} f_{i,j}(\alpha) \tag{8.13}$$

Following from Equations 8.12 and 8.13 we can express the top-k in terms of the advantage α of the protected group.

$$rND = |P - Q| \tag{8.14}$$

$$rRD(\rho) = \Big|\frac{P}{1-P} - \frac{Q}{1-Q}\Big| \tag{8.15}$$

$$skew_{G_i}@k(\rho) = log(\frac{P}{Q}) \tag{8.16}$$

$$rKL = P * ln\Big(\frac{P}{Q}\Big) - (1-P) * ln\Big(\frac{1-P}{1-Q}\Big) \tag{8.17}$$

Exposure metrics. The exposure metrics are calculated using $\rho(x)$, i.e. the ranking of individual candidates, along with a discounting function $v(k)$. When considering distributions over rankings, the candidates no longer have only one rank - instead R gives a probability of assignment in a position. Therefore we replace $\rho(x)$ with its expected value $\mathbb{E}(\rho(x))$. Thus, we can define the exposure metrics in terms of distributions of

rankings as follows:

$$\begin{aligned} exp_{G_j}(\rho) &= \frac{1}{|G_j|} \sum_{x_j \in G_j} \mathbb{E}(\rho(x_j)) \\ &= \frac{1}{|G_j|} \sum_{x_j \in G_j} v(i) \sum_{i=1}^{N} i \cdot f_{i,j}(\alpha) \end{aligned} \tag{8.18}$$

Pairwise metrics. To represent the pairwise *Rpar* metric, we observe that the likelihood that $x_i \prec x_j$ in expectation is given by the difference in the expected rank position for each candidate x_i, x_j where:

$$\mathbb{E}(\rho(x_j)) = \sum_{i=1}^{N} i \; f_{i,j}(\alpha) \tag{8.19}$$

To compute the expected *Rpar* value we then take the difference of the sign of the expected values of the position of each pair of a protected candidate and a non-protected candidate, where:

$$Rpar = \frac{1}{N} \left| \sum_{x_i \in G_p} \sum_{x_j \notin G_p} sign(\mathbb{E}(\rho(x_i)) - \mathbb{E}(\rho(x_j))) \right| \tag{8.20}$$

8.3 Metric Comparison

Advantage Functions. With our comparison framework in place, we now begin our analysis by defining a family of advantage functions f according to a set of simple and reasonable assumptions for group advantage in rankings:

$$f_{i,j}(\alpha) \geq f_{i+1,j}(\alpha) \; \forall x_j \in G_p \quad \text{if } \alpha \geq \frac{|G_p|}{N} \tag{8.21}$$

$$f_{i,j}(\alpha) \leq f_{i+1,j}(\alpha) \; \forall x_j \in G_p \quad \text{if } \alpha \leq \frac{|G_p|}{N} \tag{8.22}$$

$$\frac{1}{k} \sum_{i=1, j \in A}^{k=min(|G_i|)} f_{i,j}(\alpha) = \alpha \tag{8.23}$$

These assumptions describe an intuitive notion of group advantage controlling the distribution of groups throughout a ranking. Figure 8.2 illustrates this scenario for rankings

with 20% of the candidates in the protected group. Assumption 8.21 states that if α is greater than the overall probability of observing the protected group, then candidates in G_p will have a higher probability of being assigned favorable positions toward the top of the ranking, with uniformly decreasing probability for the lower positions (i.e. G_p has an advantage over other groups). We can see this case when $\alpha > 0.2$ in Figure 8.2. Assumption 8.22 conversely states that if the advantage is less than this value, then G_p is uniformly more likely to be observed as you move down the ranking (i.e. there is a protected group disadvantage).

Together these assumptions imply that if α equals the probability of observing the protected group over the whole ranking, then all candidates in the ranking are equally likely to be assigned to any position. Figure 8.2 illustrates this case when $\alpha = 0.2$. This scenario aligns with the definition of statistical parity wherein the advantage given to the protected group is proportional to the size of the group in the overall population. Therefore, over many rankings we would expect our fairness metrics to deem such an R as fair on average.

Finally, constraint 8.23 describes the impact of the size of the groups on f. It says that as long as there are candidates from both groups available, on average f will equal α. That is, if G_p has an advantage of $\alpha = 0.8$, then on average 80% of the candidates in top-k prefixes of the ranking will be protected candidates. However, at some rank position k (determined by the size of the smallest group) there is a tipping point where all the candidates in G_p may have been ranked, and therefore they have a lesser chance of appearing.

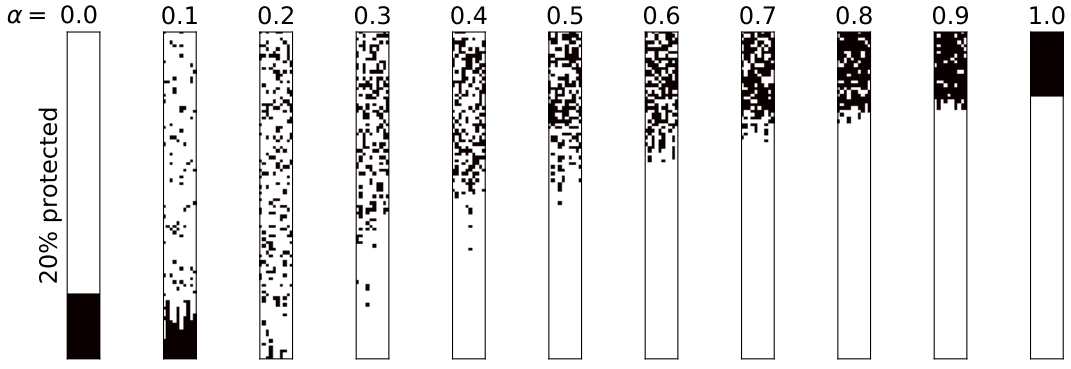


Figure 8.2: Sets of 10 rankings with 20% protected candidates with different degrees of advantage.

Characterizing Metric Behavior. From just these intuitive assumptions, we can fully characterize the behavior of the fairness metrics with respect to group advantage in the following theorems.

Theorem 8.1. *Given a ranking ρ with a protected group of candidates G_p and associated advantage α, if the assumptions in Equations 8.21 - 8.23 hold, then the rND, rRD, rKL, $expDCG$, $expRR$, and $Rpar$ metrics share the same minima.*

Proof: By definition, each of rKL, rRD, rND, $expDCG$, $expRR$ and $Rpar$ equal 0 when $P = Q$. From our advantage assumptions 8.21, 8.22, and 8.23, if $\alpha = Q = Pr(x \in G_p)$ then $P = Pr(x \in G_p \mid \rho(x) \leq k) = \frac{1}{k}\sum_{i=1}^{k}\sum_{j \in G_p} f_{i,j}(Q) = Q$ and thus each of the aforementioned metrics equals 0 when $\alpha = Q$.

Furthermore, if $\alpha > Q$ then $\frac{1}{k}\sum_{i=1}^{k}\sum_{j \in G_p} f_{i,j}(\alpha) \geq Q \ \forall \ k$ (from 8.21 and 8.23) and is strictly greater than Q for some k (from 8.23). Conversely, if $\alpha < Q$ then $\frac{1}{k}\sum_{i=1}^{k}\sum_{j \in G_p} f_{i,j}(\alpha) \leq Q \ \forall \ k$ (from 8.21 and 8.23) and is strictly less than Q for some k (from 8.23). As each of the metrics is greater than 0 when $P \neq Q$, then when considering a sum over all k the minimum will occur only at $\alpha = Q$. ■

Theorem 8.2. *Given a ranking ρ with a protected group of candidates G_p and associated advantage α, if the assumptions in Equations 8.21 - 8.23 hold, then the signs of the derivative with respect to α of the rND, rRD, rKL, and the exposure metrics are the same.*

Proof: We now show that the slopes of each of the aforementioned metrics are the same everywhere other than the critical points (i.e. at the minimum value and at the limits of the domain of α, where the derivative is undefined) when the metrics are expressed as functions of α. We accomplish this by showing that for each metric M:

$$sign(\frac{d}{d\alpha}M) = sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j \in G_p} f_{i,j}(\alpha)) \text{ if } \alpha > Q$$

$$sign(\frac{d}{d\alpha}M) = -1 \cdot sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j \in G_p} f_{i,j}(\alpha)) \text{ if } \alpha < Q$$

rKL:

$$\begin{aligned}\frac{d}{d\alpha}rKL =&(ln(\frac{P}{Q})+1)\cdot\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)\\ &+(-ln(\frac{1-P}{1-Q})-1)\cdot\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)\\ =&(ln(\frac{P}{Q})-ln(\frac{1-P}{1-Q}))\cdot\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)\end{aligned}$$

If $\alpha > Q$ then $P > Q$ then $ln(\frac{P}{Q}) - ln(\frac{1-P}{1-Q}) > 0$. Conversely, if $\alpha < 0$ then $ln(\frac{P}{Q}) - ln(\frac{1-P}{1-Q}) < 0$. Thus,

$$\begin{aligned}sign(\frac{d}{d\alpha}rKL) =&sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)) \text{ if } \alpha > Q\\ sign(\frac{d}{d\alpha}rKL) =&-1\cdot sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)) \text{ if } \alpha < Q\end{aligned}$$

rND:

$$\frac{d}{d\alpha}rND =\frac{P-Q}{|P-Q|}\cdot\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)$$

As $\frac{P-Q}{|P-Q|} > 1$ when $\alpha > Q$ and $v\frac{P-Q}{|P-Q|} < 1$ when $\alpha < Q$,

$$\begin{aligned}sign(\frac{d}{d\alpha}rND) =&sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)) \text{ if } \alpha > Q\\ sign(\frac{d}{d\alpha}rND) =&-1\cdot sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)) \text{ if } \alpha < Q\end{aligned}$$

rRD:

$$\frac{d}{d\alpha}rRD = \frac{\left(\frac{x}{(1-x)^2}+\frac{1}{1-x}\right)\left(\frac{x}{1-x}-\frac{Q}{1-Q}\right)}{\left|\frac{x}{1-x}-\frac{Q}{1-Q}\right|}\cdot\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)$$

When $\alpha > Q$ then $\left(\frac{x}{1-x}-\frac{Q}{1-Q}\right) > 0$, which implies $\frac{\left(\frac{x}{(1-x)^2}+\frac{1}{1-x}\right)\left(\frac{x}{1-x}-\frac{Q}{1-Q}\right)}{\left|\frac{x}{1-x}-\frac{Q}{1-Q}\right|} > 0$. Conversely, if $\alpha < Q$ then $\frac{\left(\frac{x}{(1-x)^2}+\frac{1}{1-x}\right)\left(\frac{x}{1-x}-\frac{Q}{1-Q}\right)}{\left|\frac{x}{1-x}-\frac{Q}{1-Q}\right|} < 0$. Thus,

$$sign(\frac{d}{d\alpha}rRD) = sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)) \text{ if } \alpha > Q$$

$$sign(\frac{d}{d\alpha}rRD) = -1\cdot sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p}f_{i,j}(\alpha)) \text{ if } \alpha < Q$$

Exposure:

$$\begin{aligned}\frac{d}{d\alpha}exp_{G_p}(\rho) &= \frac{d}{d\alpha}\Big|\frac{1}{|G_p|}\sum_{i=1}^{k}\sum_{j\in G_p}v_i\cdot i\cdot f_{i,j}(\alpha)\\ &\quad -(1-\frac{1}{|G_p|}\sum_{i=1}^{k}\sum_{j\in G_p}v_i\cdot i\cdot f_{i,j}(\alpha))\Big|\\ &= \frac{d}{d\alpha}\Big|\frac{2}{G_p}\sum_{i=1}^{k}\sum_{j\in G_p}v_i\cdot i\cdot f_{i,j}(\alpha)-1\Big|\\ &= \frac{\frac{2}{G_p}\sum_{i=1}^{k}\sum_{j\in G_p}v_i\cdot i\cdot f_{i,j}(\alpha)-1}{|\frac{2}{G_p}\sum_{i=1}^{k}\sum_{j\in G_p}v_i\cdot i\cdot f_{i,j}(\alpha)-1|}\\ &\quad\cdot\frac{d}{d\alpha}\frac{2}{G_p}\sum_{i=1}^{k}\sum_{j\in G_p}v_i\cdot i\cdot f_{i,j}(\alpha)\end{aligned}$$

If $\alpha < q$, then the term that the above derivative is multiplied by is less than 0, and if $\alpha > Q$ then the term is positive. Additionally, for v_i as defined for both *expDCG* and

expRR the following holds:

$$sign(\frac{d}{d\alpha}\frac{2}{G_p}\sum_{i=1}^{k}\sum_{j\in G_p} v_i \cdot i \cdot f_{i,j}(\alpha)) = sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p} f_{i,j}(\alpha))$$

Thus,

$$sign(\frac{d}{d\alpha}exp_{G_p}) = sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p} f_{i,j}(\alpha)) \text{ if } \alpha > Q$$

$$sign(\frac{d}{d\alpha}exp_{G_p}) = -1 \cdot sign(\frac{d}{d\alpha}\frac{1}{k}\sum_{i=1}^{k}\sum_{j\in G_p} f_{i,j}(\alpha)) \text{ if } \alpha < Q$$

We have thus shown that if the assumptions in Equations 8.21 - 8.23 hold, then signs of the derivative with respect to α of the rND, rKL, $expDCG$, $expRR$, and metrics are the same.

■

Theorem 8.3. *Given a ranking ρ with a protected group of candidates G_p, $Rpar(\rho)$ has its maximum value when $\alpha = 0$ or $\alpha = 1$, meaning one group has a total advantage.*

Proof:

If either group has total advantage, then $sign(\mathbb{E}(\rho(x_i)) - \mathbb{E}(\rho(x_j))) = sign(\mathbb{E}(\rho(x_m)) - \mathbb{E}(\rho(x_n)))$ for all $x_i, x_m \in G_p$ and all $x_j, x_n \notin G_p$. Thus, $Rpar = \frac{1}{N}\Big|\sum_{x_i\in G_p}\sum_{x_j\notin G_p} sign(\mathbb{E}(\rho(x_i)) - \mathbb{E}(\rho(x_j)))\Big| = 1$. This is the maximum *Rpar* value, because
$sign(\mathbb{E}(\rho(x_i)) - \mathbb{E}(\rho(x_j))) \neq sign(\mathbb{E}(\rho(x_m)) - \mathbb{E}(\rho(x_n))) \implies$
$\frac{1}{N}\Big|\sum_{x_i\in G_p}\sum_{x_j\notin G_p} sign(\mathbb{E}(\rho(x_i)) - \mathbb{E}(\rho(x_j)))\Big| < 1.$ ■

8.3.1 Key Metric Comparison Observations

It follows from Theorem 8.1 that **optimizing for any of these core metrics optimizes for all.** Theorem 8.2 additionally shows that if we improve fairness according to one of the metrics, **we improve the others as well**. We do not include the *Rpar* metric in this analysis since *Rpar* is computed using the discrete set of possible pairs in the ranking, and therefore the function has a non-continuous range. However we observe in our empirical evaluation in Section 8.5 that the pairwise metric does indeed exhibit similar behavior to the rest of the metrics. Together this comparative analysis shows that when group advantage can be expected to conform to the assumptions laid out,

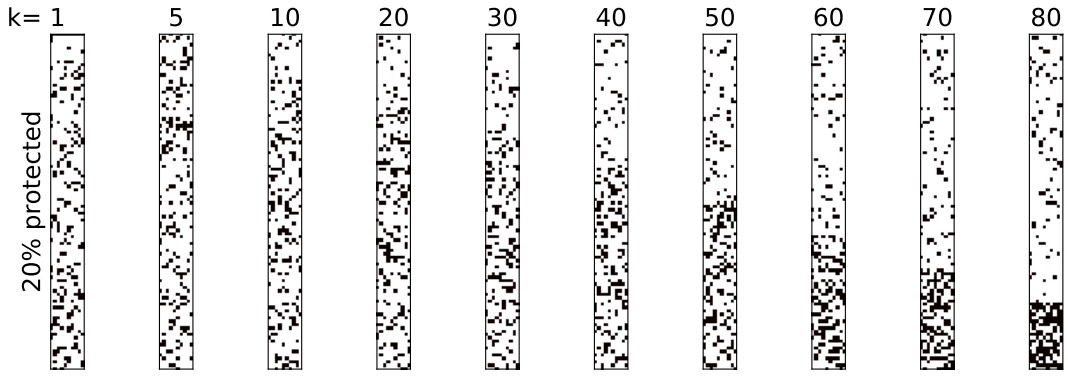
any choice of evaluation metric is appropriate when the goal is to optimize for as fair a ranking as possible.

Additionally, we observe that the metrics do not behave the same in terms of their maximum values. We prove in Theorem 8.3 that the pairwise *Rpar* metric has a maximum value in the case where either group **has a total advantage over the other.** This matches with one intuitive notion of *unfairness* wherein a total group advantage is as extreme a violation of statistical parity possible. On the other hand, as we observe empirically in Section 8.5, the maximum values of other metrics depend on the size of the groups. If one group has a strong majority and total advantage, they are assigned a more fair score than if a minority group has a total advantage. One reason for this could be that if the items are ranked at random, a group with many more candidates is more likely to have gotten an advantage by chance. However this also might be a primary case when fairness evaluation is needed - when one group suffers from disparate representation in the dataset. In fact, most of the metrics we consider **will not flag rankings as unfair which strongly disadvantage a minority group**. These competing notions of what constitutes unfairness deserve careful consideration when selecting a fairness metric in an applied setting.

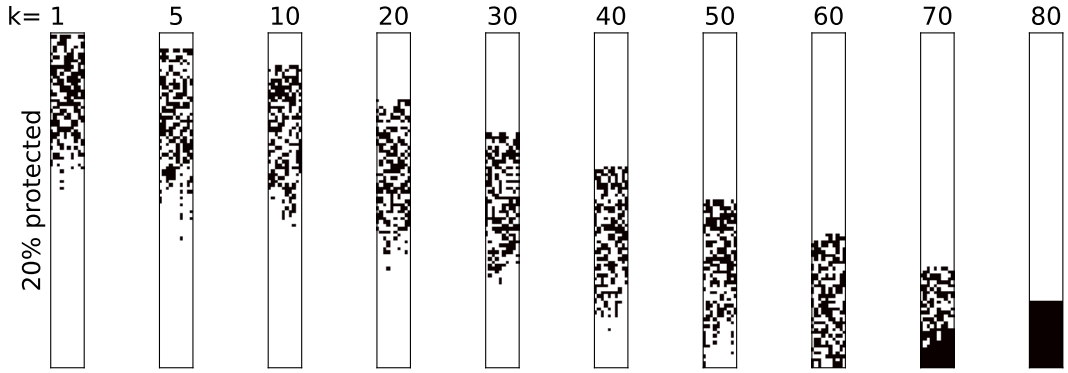
8.4 Alternative Advantage Functions

In our analysis so far, we consider functions of advantage f which are applied *smoothly* throughout the ranking. However, in the real-world, it may be that other factors impact the probability of candidates being assigned to positions. If we relax our assumptions about f decreasing uniformly through the ranking, we can imagine scenarios where bias may fluctuate throughout the ranking.

Rooney Rule. As an example, we consider the Rooney rule for hiring which has been a topic of interest by the fairness community [33, 97]. This strategy dictates that at least one candidate from the protected group is included in the top-k positions in the ranking. This is meant as a fairness-correcting intervention to ensure representation of the protected group. However one can also imagine an unfair manipulation using a similar strategy to ensure that a candidate from the advantaged group is always guaranteed the top spot in the ranking. Figure 8.3 illustrates versions of these scenarios. Sets of 10 randomly generated rankings are shown for a dataset with 20% protected candidates. In Figure 8.3a, for different values of k, the rankings are required to assign 10% of the top-k rank positions to the protected group. Below k, the rank positions are assigned



(a) Top-k positions are assigned 10% to candidates from the protected group, while the rest of the ranking is randomly assigned.



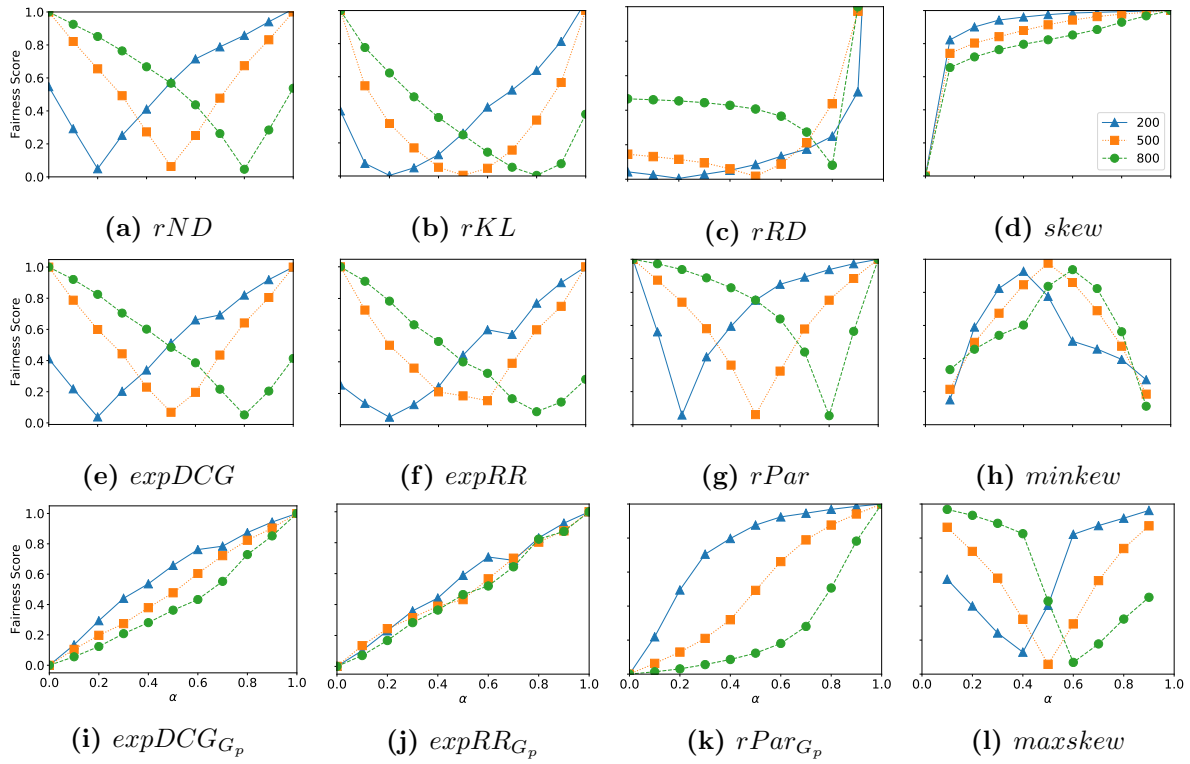
(b) Top-k positions are reserved for candidates from the non-protected group. In the rest of the ranking the protected group is advantaged by $\alpha = 0.5$.

Figure 8.3: Sets of 10 random rankings with 20% protected candidates generated using alternative advantage functions.

randomly. Figure 8.3b illustrates another scenario where the top-k positions are all reserved for candidates from the non-protected group, and then in the rest of the ranking the protected group is advantaged by $\alpha = 0.5$.

For these rankings, it is clear that unfairness manifests in different ways than in our previous analysis. In Figure 8.3a, if k is small then the rankings overall will align with our notion of statistical parity, however if k is very large then the protected group is actually disadvantaged. Figure 8.3b clearly depicts rankings that are unbalanced, however one could imagine that the unfairness at the top and bottom of the ranking may cancel each other out. Comparing these examples to our original rankings in Figure 8.2 we can see that the advantage of each group is more subtly distributed throughout the rankings in this case.

8.5 Evaluation



(a) rND **(b)** rKL **(c)** rRD **(d)** $skew$

(e) $expDCG$ **(f)** $expRR$ **(g)** $rPar$ **(h)** $minkew$

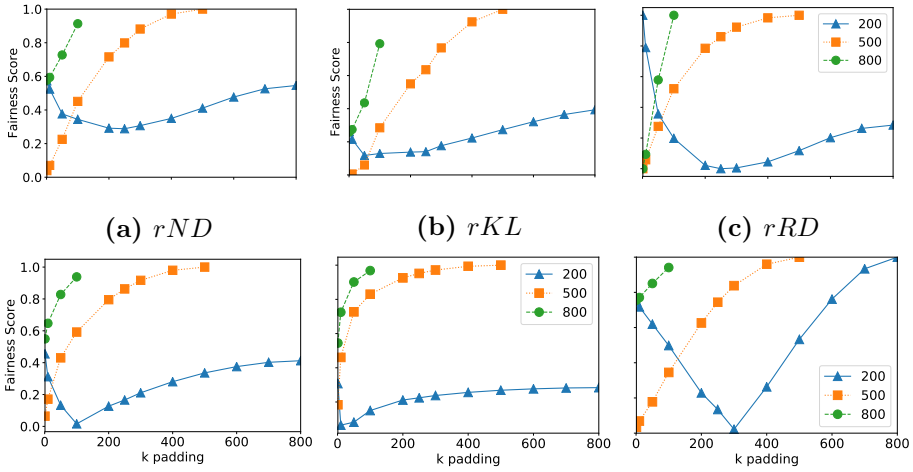
(i) $expDCG_{G_p}$ **(j)** $expRR_{G_p}$ **(k)** $rPar_{G_p}$ **(l)** $maxskew$

Figure 8.4: Fairness metrics applied to rankings with different α values using a smooth function of advantage.

To gain a tangible understanding of the performance of fairness metrics for ranking, we now present an empirical evaluation of evaluation measures.

Data Generation. We produce random rankings using different functions of group advantage. For each experiment results are averaged over 10 runs. We consider rankings where the protected group is the minority (20% protected), where the groups are balanced (50% in each group), and where the protected group is the majority (80% protected). Our standard group advantage in Section 8.3 is generated following the algorithm given by Yang and Stoyanovich [155]. We then adapt this strategy to produce the alternative advantage functions presented in Section 8.4.

Metrics. We evaluate all metrics included in our initial correlation analysis in Section 8.1.4: top-k metrics rND, rRD, rKL, $skew$, $minskew$ and $maxskew$, pairwise $Rpar$ and exposure based $expDCG$ and $expRR$. Again we also consider pairwise and exposure values only for the protected group as $Rpar_{G_p}$, $expDCG_{G_p}$, and $expRR_{G_p}$. Metric values are evaluated for varying levels of advantage $\alpha \in [0.0, 1.0]$. To facilitate comparison, we



(a) rND **(b)** rKL **(c)** rRD

(d) $expDCG$ **(e)** $expRR$ **(f)** $Rpar$

Figure 8.5: Fairness metrics applied to rankings where the top-k positions are reserved for candidates from the non-protected group, for different values of k. In the rest of the ranking the protected group is advantaged by $\alpha = 0.5$.

normalize each metric to lie in a range of zero to one (scaling based on the minimum and maximum possible values given the size of the groups). We note that for some metrics the maximum values are unbounded, and we therefore omit some extreme values.

Standard Assumption of Advantage. Figure 8.4 compares the behavior of all the metrics across different α values of advantage, for rankings with different size groups. There are clearly observable similar patterns for our core metrics rND, rKL, $Rpar$, $expDCG$ and $expRR$. The rRD metric on the other hand exhibits different behavior. The minima are the same as the other metrics, but when the protected group is totally favored, the rRD metric explodes, skewing the results. We have scaled the figure based on the max value for $\alpha = 0.9$ for readability. The rest of the metrics considered in our correlation analysis again can be observed to follow different patterns with respect to group advantage. The $skew$, $Rpar_{G_p}$, $expDCG_{G_p}$, and $expRR_{G_p}$ metrics indeed capture only half the story. The $maxskew$ and $minskew$ metrics better align with our core group, however when either of the groups has a total advantage these values explode. We omitted these values for readability.

Extreme Advantage. One key difference among the core metrics can be observed in Figure 8.4. Aligning with our analysis in Theorem 8.3, we see that only the pairwise $Rpar$ metric always assigns a maximum unfair score in both extreme cases where one group is completely advantaged over the other. The $expDCG$, rND, and rKL metrics give a more fair score when the majority group is totally favored over the minority group.

Alternative Advantage Functions. Finally we highlight an example to show that under relaxed assumptions, the metrics are no longer guaranteed to follow the same patterns. We consider the alternative advantage scenario described in Section 8.4 and shown in Figure 8.3b. Here the k top spots in each ranking are given to candidates from the non-protected group. After this top-k padding, the protected group is then given an advantage of $\alpha = 0.5$. In figure 8.5 k is varied along the x axis of the charts, showing that when k is small (meaning the protected group is concentrated toward the top of the ranking) the metrics totally disagree on whether this is fair or unfair, in particular when the protected group is the minority. The *Rpar* metric indicates very unfair around 0.8 and *expRR* gives a score close to 0. The majority of the metric values stay relatively flat for the 20% protected case as well, while *Rpar* goes through the range of all possible values. At a certain point, the pairwise advantage between the groups balances out and appears fair.

Indeed, this manifestation of group advantage is unusual. It may even be hard even for a human analyst to decide whether the ranking is fair or unfair. This is a good example of the nuanced determination required for assessing fairness. Without an understanding of how these fairness metrics behave, ranking systems may give unexpected behavior.

9

Fair Rank Aggregation

To date, efforts for fair ranking have been limited in that they only consider fairness of a *single* ranking in isolation. The critical yet so far overlooked problem of *ranking by consensus* arises when numerous decision makers produce rankings over candidate items, and then those rankings are *aggregated* to create a final *consensus ranking* [24, 93]. While many (non-fairness aware) procedures for aggregating a set of rankings have been put forth by the database community [5, 23, 63, 89], the problem of *fair rank aggregation* remains open. It is largely unexplored whether aggregation might introduce or exacerbate bias disadvantaging particular groups. In this Chapter we thus set out to investigate these open problems. This work is in submission to a major conference in 2020 as the paper:

> Caitlin Kuhlman and Elke A. Rundensteiner,
> *Rank Aggregation Algorithms for Fair Consensus.*

9.1 Introduction

9.1.1 Hiring Example.

Consider the university hiring scenario in Figure 9.1. After reviewing a pool of faculty applicants (assisted by an automated screening tool [150]), each committee member ranks the candidates based on their individual impressions. Now the committee must come to some overall *consensus ranking* to recommend to their department. As seen in Figure 9.1, this poses several challenges. First, the procedure to combine the rankings is not obvious. For instance, candidate A is most frequently ranked in the top spot, but also seems to be divisive, since two committee members ranked A last. Candidate B on

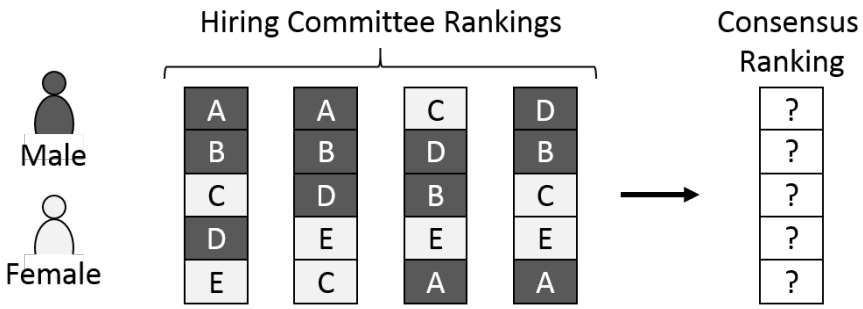


Figure 9.1: Hiring committee rankings to be aggregated. Four committee members each rank the set of candidates $\{A, B, C, D\}$ from two groups based on gender.

the other hand is consistently ranked near the top, but never in the number one spot. Another consideration is the committee's desire to have a diverse faculty body. They would like a gender balance among the candidates in the final ranking to compensate for any unintended bias in the input rankings where female candidates seem to be ranked lower than males. Clearly, a principled strategy is required to fairly account for this imbalance while still appropriately representing all committee members' preferences.

9.1.2 State-of-the-Art

Contemporary (group) fairness research [65] is concerned with predictive outcomes for minority or otherwise disadvantaged groups defined by sensitive legally protected data attributes such as race, gender, or age. As discussed in Chapter 8, the bulk of recent research for ranking targets a *statistical parity* notion of fairness [10, 34, 70, 106, 155, 161], aiming to ensure that each group of *candidates* being ranked receives a proportionate number of preferred rank positions. These methods consider fairness only in the context of *one single* ranking. To the best of our knowledge, *group fairness has not yet been explored when aggregating multiple rankings.*

Traditional rank aggregation produces an aggregate ranking (a consensus) by finding the ranking *closest* to the base set of input rankings [93]. This task, having broad applications, has seen much interest in the database [24], machine learning [110], and information retrieval [60] fields. Many algorithms have been proposed for finding optimal or approximate consensus [13, 41, 43, 60, 93, 120, 138, 157]. Considering each ranking as representing the preferences of a *voter* for a given candidate set, Social Choice Theory ensures each voter has an equal say [9]. While properties of aggregation algorithms with regard to the voters are well understood [24], fair and equitable treatment of the

candidates being ranked in the context of rank aggregation remains unaddressed.

9.1.3 Challenges

To address this open problem of group fairness for rank aggregation, the following challenges must be tackled.

Aggregation problem complexity. Classical rank aggregation – even without considering fairness – is a hard problem. Depending on the criteria used to determine the consensus ranking, finding the exact solution may be NP-hard [13, 60]. For a given set of base rankings over n candidates, there are $n!$ possible consensus rankings to chose from, making an exhaustive search over all options intractable. The complexity of the aggregation problem is not only impacted by the number of candidates being ranked, but also by the number of voters (base rankings), and the extent to which the base rankings agree (or disagree) on the placement of individual candidates [5]. This becomes further complicated for fair aggregation, since bias in the individual base rankings could also impact the complexity of the task.

Notions of fair aggregation. Rankings may be produced according to heterogeneous schemes, including human decision makers' preferences or proprietary ranking algorithms. Therefore, fair aggregation must be performed without access to the underlying data and ranking models – rendering causal notions of fairness [135, 153] impossible, as we cannot investigate the relationship between data attributes and outcomes. Associational bias mitigation methods exist for single rankings [10, 34, 70, 155, 161], which typically trade accuracy of the rank order for fairness. However it is unclear how the similarities and differences among a set of rankings relate to these ways of imposing fairness. Therefore how best to incorporate such measures into the rank aggregation problem is an open question.

Competing optimization objectives for fair aggregation. If unfair bias against some group is present in base rankings, it is not known how aggregating them into a single ranking will impact this bias. Perhaps aggregation may exaggerate a slight advantage for one group creating a more pronounced bias in the final consensus. This could demand a high toll in aggregation accuracy to ensure fairness. Conversely, a diversity of perspectives among the voters (base rankings) might inherently mitigate unfairness present in only a few of the rankings, in which case correction is not necessary. A deep study of these subtle inter-dependencies is needed. Beyond that, a sophisticated strategy is required capable of balancing the competing goals of fairness for the groups of candidates being

ranked while concurrently retaining a good representation of the base rankings.

9.1.4 Proposed Approach

To address these challenges, we formalize fair rank aggregation as a constrained optimization problem balancing the competing objectives above. The solution is defined as the closest consensus ranking to the base set of rankings that satisfies a targeted fairness criterion. We then propose a which uses pairwise discordance to both compute closeness among consensus and base rankings and measure the advantage given to each group of candidates. This allows group fairness criteria to be seamlessly integrated into Kemeny optimal rank aggregation [93]. We demonstrate the power of our pairwise framework to support a *pairwise rank parity* fairness definition [106], proving an equivalence between our approach and popular top-k statistical parity [155] metrics for fair ranking.

Next we leverage the pairwise framing of the problem to tackle the complexity challenge of fair rank aggregation. As first solution, we propose an integer linear program with parity constraint to produce a Kemeny-optimal fair consensus. This approach can aggregate many rankings generated by a large number of *voters*. However, the large number of binary variables is prohibitive when there are many *candidates*. Therefore we extend this by deriving a lower bound on the cost of pairwise fairness criteria. This supports the design of a rank parity-preserving search heuristic integrated into a branch-and-bound fair rank algorithm, which we call Fair-BB. We demonstrate that Fair-BB speeds up computation when ranking many *candidates* when the fairness requirements are lenient. Finally, we provide a fast approximation post-processing algorithm Fair-Post which guarantees fairness while introducing minimal pairwise error, and scales to millions of candidates.

We thoroughly evaluate these alternate solution strategies in a rich test bed of rank aggregation scenarios. The previously unknown relationship between fairness and consensus among multiple rankings is explored, using the Mallows model [116] to generate distributions of rankings and expose the tradeoffs between our competing objectives. Finally, we demonstrate the ability of our framework to produce fair consensus on real-world using a case study of sports rankings. Our methods consistently produce fair aggregations, extending contemporary fairness to ranking by consensus.

Contributions of our work include:

1. We formulate the open problem of fair rank aggregation as finding consensus among a set of input rankings while ensuring the fair treatment of candidates being ranked.

2. We propose a novel for parity-preserving Kemeny aggregation.
3. We design a series of algorithms which guarantee to find optimal fair consensus, leveraging integer linear optimization and custom branch-and-bound strategies.
4. We also design a fast approximation algorithm which finds a fair solution with minimal aggregation error.
5. We study the interplay between rank consensus and group fairness, evaluating the relative performance of our solutions for a wide spectrum of aggregation scenarios.
6. We investigate unfair group bias for rankings generated by human decision makers using a real-world case study of expert rankings of sports players.

9.2 Problem Formulation

Traditional Rank Aggregation. The set of all possible rankings of X is S_n the symmetric group of permutations. In the traditional rank aggregation problem [93] we are given a subset of base rankings $R \subseteq S_n$ created by some *voters*. To be broadly applicable, we do not make assumptions about how the base rankings were determined, but rather consider R as a fixed input. We are tasked with finding a single *consensus ranking* ρ^* that best represents R as given in Definition 9.1. The consensus ranking ρ^* is the median ranking in S_n with the minimum average distance to the rankings in R according to some distance function d. A median ranking always exists, however it may not be unique.

Definition 9.1. *Given a set of base rankings $R \subseteq S_n$ and distance function d* ***the traditional rank aggregation problem*** *is to find a closest ranking $\rho^* \in S_n$ to R such that:*

$$\rho^* = \arg\min_{\rho \in S_n} \frac{1}{|R|} \sum_{\sigma \in R} d(\rho, \sigma)$$

Fair Rank Aggregation Problem Formulation. Unfair bias can be defined in different ways. Since we do not necessarily have access to the underlying data attributes or ranking procedures used by the voters, in this work we aim to meet some group fairness criterion F determined only by the rank order of the candidates and their group membership. We initially focus

We now formalize our fair aggregation problem in Definition 9.2 as the goal of finding the closest consensus ranking to R that satisfies a fairness criterion F. Henceforth we will

focus on achieving *statistical parity* among groups in the final consensus ranking ρ^*, for two groups of candidates. Our general framework can be adapted to additional fairness definitions.

Definition 9.2. *Given a set of base rankings $R \subseteq S_n$ and a fairness criterion F,* ***the fair rank aggregation problem*** *is to find a closest ranking $\rho^* \in S_n$ to the base rankings that satisfies F.*

9.3 Proposed Framework for Fair Rank Aggregation

9.3.1 Solution Spectrum for Fair Aggregation

We now examine the spectrum of alternate approaches for producing an aggregate ranking that concurrently guarantees fairness for the candidates being ranked while assuring maximal representation of the interests of the voters expressed by their respective rankings. For this, as illustrated in Figure 9.8, let us suppose we have a fairness correction method $f(\rho) = \rho'$ that can rearrange the items *in a single ranking* to satisfy a fairness criterion F (such as top-k parity) in a way that minimizes the distance to the original ranking $d(\rho, \rho')$. This spectrum then ranges from the one extreme (top of Figure 9.8) of applying fairness mitigation as pre-processing before constructing an aggregate ranking to the other extreme (middle of Figure 9.8) of applying fairness mitigation only after traditional rank aggregation as a post-processing step. As we show, neither of these extreme solutions are adequate, necessitating instead the development of a novel integrated dual-optimization strategy which optimizes for these two competing goals concurrently (bottom of Figure 9.8). These observations are also validated empirically in our experimental evaluation in Section 9.5.

The **pre-processing** strategy illustrated at the top of Figure 9.8 first applies this correction method one by one to each of the given rankings in the base set R, to make a new set of fair rankings R'. Thereafter, R' is aggregated using a traditional rank aggregation method. This approach guarantees neither optimal distance to the original base set R, nor fairness of the resulting ranking. One, f may minimize the distance between the fair rankings in R' and their original counterparts in R, however this does not necessarily imply that the median ranking for R' is close to the original R. Two, even if each ranking in R' meets the fairness criterion, the order of individual items may differ

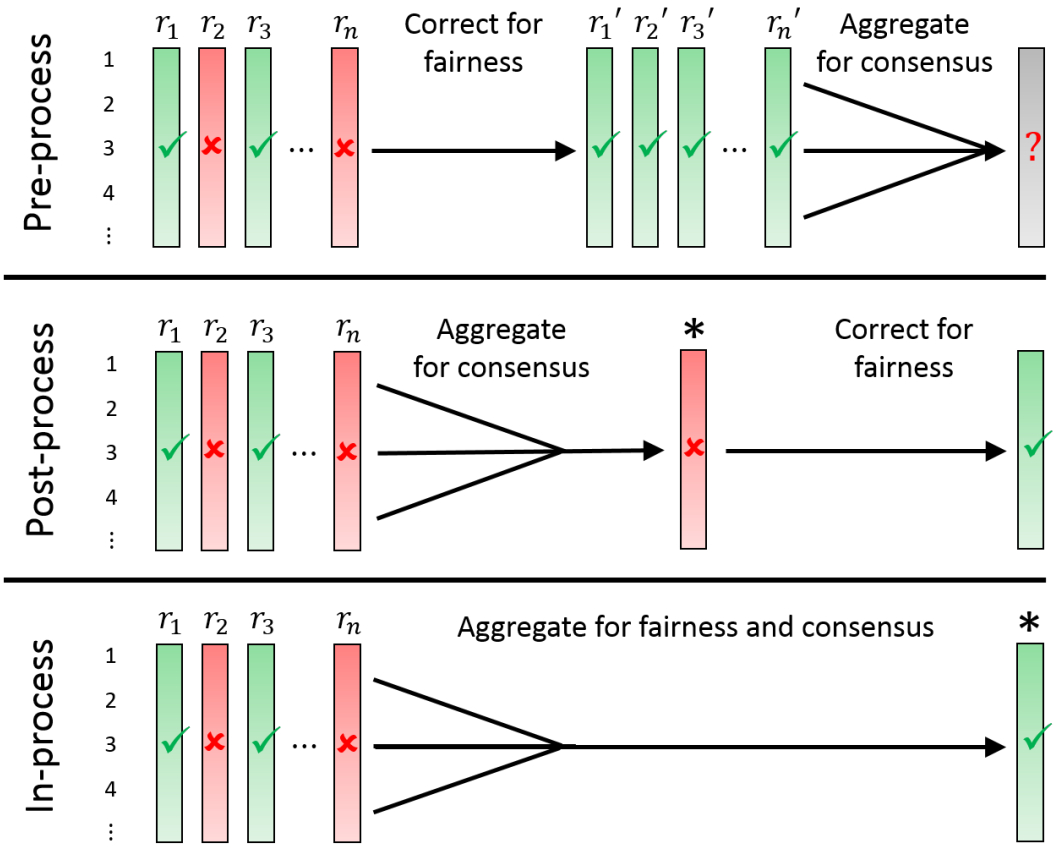


Figure 9.2: Alternative fair rank aggregation strategies.

greatly across those adjusted rankings in R'. Therefore we do not have any knowledge on whether the consensus ranking constructed by aggregating R' would also meet the fairness criterion.

On the other hand, the **post-processing** strategy shown in the middle of Figure 9.8 first aggregates R without considering fairness, producing a consensus ranking ρ^*. Then ρ^* is corrected for fairness to produce a ranking $f(\rho^*) = \rho'$. This time we can be sure ρ' is fair. However, we have no guarantee about the quality of the resulting aggregation. That is, even if the corrected consensus ranking ρ' could be guaranteed to be close to the original ρ^*, it may not be the closest fair solution to the base rankings in R. Therefore neither of these extreme approaches can guarantee both fairness and aggregation accuracy.

9.3.2 Integrated Pairwise Solution

We propose a conceptual framework for an integrated fair aggregation solution that elegantly balances aggregation accuracy with fairness, achieving both goals concurrently as shown in the bottom of Figure 9.8. Our key insight here is that we must align the measure of distance between rankings with the measure of candidate group advantage using a *pairwise rank representation*. As we will demonstrate below, this then allows for fairness criteria to be integrated seamlessly into a fair aggregation framework.

Any ranking can be represented as a series of pairwise comparisons between the candidates. Given a base set of rankings R, they will agree on the order of some pairs, and disagree on others. The number of pairs which appear in an inverted order in one ranking compared to the other corresponds to the distance measure known as the **Kendall Tau** [94], defined in Equation 9.1 for two rankings $\rho, \sigma \in S_n$.

$$K(\rho, \sigma) = \sum_{x_i, x_j \in X} \mathrm{I}\big(\rho(x_i) \prec \rho(x_j) \text{ and } \sigma(x_j) \prec \sigma(x_i)\big) \tag{9.1}$$

Here I is the indicator function, with $I(x) = 1$ when x is true, and $I(x) = 0$ otherwise. The consensus ranking with the minimum average Kendall tau distance to the rankings in R, given in Equation 9.2, is known as the **Kemeny optimal rank aggregation** [93].

$$\rho^* = \arg\min_{\rho \in S_n} \frac{1}{|R|} \sum_{\sigma \in R} K(\rho, \sigma) \tag{9.2}$$

Although there are different strategies for measuring the distance between rankings, Kemeny aggregation is seen as the gold standard for rank aggregation, which concurrently satisfies multiple axioms from Social Choice Theory [24]. For this reason, it has seen extensive database and machine learning applications [5, 23, 63, 89]. Kemeny aggregation provides our starting place from which to consider group fairness in the ranking utilizing pairs of candidates.

Consider the case where our database X contains candidates from two different groups, G_1 and G_2. The pairs in the Cartesian product X^2 can be divided into three subsets: those pairs containing only candidates from group G_1, those containing only candidates from G_2, and the set of "mixed" pairs containing one item from each group. In Chapter 7, we proposed that the proportion of the heterogeneous pairs which favor one group over the other corresponds to a fairness measure of the relative advantage that group enjoys in the ranking [106]. We now observe that $Rpar_{G_1}$ computes the probability

that an item from group G_1 is ranked above an item from group G_2, such that:

$$Rpar_{G_1}(\rho) = P(x_i \prec_\rho x_j \mid x_i \in G_1, x_j \in G_2) \tag{9.3}$$

Following from this, we propose to adopt the pairwise formulation of statistical parity in Definition 9.4, Given a ranking $\rho = [x_1 \prec x_2 \prec ... \prec x_n]$ of candidates belonging to mutually exclusive groups G_1, G_2, ρ satisfies pairwise statistical parity for two groups is met if the following condition is met:

$$Rpar_{G_1}(\rho) = Rpar_{G_2}(\rho) \tag{9.4}$$

9.3.3 Equivalence between Top-k and Pairwise Statistical Parity

Defining parity this way gives a compatible fairness criterion for Kemeny aggregation, in that both optimization goals now depend on the pairwise ordering of candidates in the consensus ranking. We know from the comparative analysis of statistical parity metrics in Chapter 8 that pairwise metrics behave similarly to top-k in the presence of different manifestations of group advantage in a single ranking. We now further show how the pairwise Definition 9.4 relates to the notions of statistical parity expressed in Definitions 8.1 and 8.2. The pairwise formulation indeed corresponds to a variant of top-k parity semantics, namely, a summary formulation that takes all possible prefixes of the list into account. For this, we observe that the probability on the left hand side of the Top-k Parity in Definition 8.2 corresponds to the cumulative distribution function for the rank $\rho(x)$ of any $x \in G_1$. Let us denote this as $F_{\rho(x)|G_1}(k)$ such that:

$$F_{\rho(x)|G_1}(k) = P(\rho(x) \leq k \mid x \in G_i) \tag{9.5}$$

To compute the value of $F_{\rho(x)|G_1}$ for a given k, we can simply count the proportion of candidates from group G_1 in the top-k prefix of the ranking as per below.

$$F_{\rho(x)|G_1}(k) = \sum_{x \in G_1} \frac{I(\rho(x) \leq k)}{|G_1|} \tag{9.6}$$

Since the rank of x is strictly positive, we can use
$F_{\rho(x)|G_1}(k)$ to compute the conditional expectation of the rank of x given membership in

G_1, where

$$E(P(\rho(x) = k \mid x \in G_1) = \sum_{k=0}^{\infty} \Big(1 - \sum_{x \in G_1} \frac{I(\rho(x) \leq k)}{|G_1|}\Big) \tag{9.7}$$

Each possible value of k corresponds to a rank position in ρ assigned to some candidate x. Let K_1 and K_2 be the set of all rank positions assigned to candidates in G_1 and in G_2, respectively. We can then re-write Equation 9.7 as:

$$\begin{aligned} E(P(\rho(x) = k \mid x \in G_1) = \sum_{k=0}^{n} 1 - \Big(& \sum_{x \in G_1} \sum_{k_2 \in K_2} \frac{I(\rho(x) \leq k)}{|G_1|} \\ & + \sum_{x \in G_1} \sum_{k_1 \in K_1} \frac{I(\rho(x) \leq h)}{|G_1|} \Big) \end{aligned} \tag{9.8}$$

The first outer term in Equation 9.8 is simply $n+1$, since for any $k > n$ the probability that $\rho(x) \leq k$ is always 1. The inner summations describe pairwise relationships between the candidates in ρ. The first term is only over those candidates $x \in G_1$ ranked higher than candidates in G_2 (each position $k_2 \in K_2$ corresponding to a candidate in G_2). These are the same pairs used to compute $Rpar_{G_1}$. The second term is only over candidates in G_1. For each consecutive position k_1, this term counts the candidates ranked at that position or higher. This thus corresponds simply to the sum of consecutive integers from 1 to $|G_1|$, given by the constant $\frac{|G_1|(|G_1|+1)}{2}$. Therefore we have:

$$E(P(\rho(x) = k \mid x \in G_1) = n + 1 - \Big(|G_2|\ Rpar_{G_1} + \frac{|G_1| + 1}{2}\Big) \tag{9.9}$$

From Equation 9.9, we can observe two things. One, the top-k probability taken over all k is a linear function of the *Rpar* measure, where:

$$\sum_{k=0}^{n} P(\rho(x) \leq k \mid x \in G_1) = |G_2|\ Rpar_{G_1} + \frac{|G_1| + 1}{2} \tag{9.10}$$

Two, *Rpar* can be used to measure the difference in the expected rank position of different groups. This is not surprising, as we further observe that the *Rpar* score for a disadvantaged group is equivalent to the Mann–Whitney U-statistic [118], where $Rpar = \frac{U}{|G_2||G_1|}$. The Mann-Whitney U has a well-known relationship to the area under the ROC curve for a probabilistic classifier [77]. Namely, given true positive instances TP and true negative instances TN, the area under the ROC curve (AUC) measures the probability that a true positive is ranked above a true negative according to the predicted likelihood

of membership in the positive class [77] This is determined by the change in the true positive rate with respect to the false positive rate, such that

$$AUC = \int_{t=-\infty}^{\infty} P(\hat{y} \geq t | x \in TP)\ dP(\hat{y} \geq t | x \in TN) = \frac{U}{|TP||TN|}$$

where t is the classification threshold. Similarly, we put forth here that the *Rpar* score relates to the rates of positive outcomes achieved by each group according to different values of k where:

$$Rpar_{G1} = \int_{k=1}^{n} P(\rho(x) \leq k \mid x \in G_1)\ dP(\rho(x) \leq k \mid x \in G_2) \tag{9.11}$$

Rank Parity for Fair Hiring Example. We return to our example from Section 9.1.1 to study how imposing pairwise statistical parity on the consensus ranking ρ^* will impact rank aggregation. Figure 9.3 illustrates the result of Kemeny aggregation with and without rank parity. Rank 9.3a is the baseline unconstrained Kemeny ranking. As we see, it reflects the tendency of female candidates to be ranked lower than males, and it places the controversial candidate A at the top. Five mixed pairs favor males over females $\{(a \prec c), (a \prec e), (b \prec c), (b \prec e), (c \prec d)\}$ and only one pair favors females over males $\{(c \prec d)\}$. In contrast, ranking 9.3b overcomes this group disparity. In fact, it is the closest ranking to the base set which also satisfies parity. In this ranking, females enjoy equal pairwise advantage to males, where three pairs favor males $\{(b \prec c), (b \prec e), (d \prec e)\}$ and three favor females $\{(c \prec d), (c \prec a), (e \prec a)\}$.

9.4 Proposed Methods for Kemeny-optimal Fair Aggregation

Given our integrative pairwise framework, we now design optimal strategies to solve the proposed constrained optimization problem for fairness-preserving rank aggregation. To facilitate the design of our methods, we employ a compact representation of the rankings in R in the form of the *precedence matrix* C (Def. 9.3). Matrix C summarizes the pairwise relationships between candidates in X.

Definition 9.3. *Given a set of rankings to be aggregated R over a dataset $X = \{x_1, \ldots, x_n\}$,*

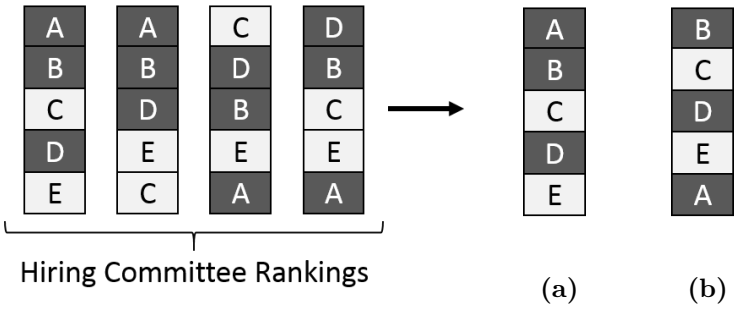


Figure 9.3: Aggregated hiring committee rankings with and without fairness criteria, namely, Kemeny optimal ranking (a) without considering fairness, and (b) with rank parity.

then the **precedence matrix** C *is defined as:*

$$C_{ij} = \frac{1}{|R|} \sum_{\sigma \in R} I(x_i \prec_\sigma x_j)$$

Each entry in C indicates the proportion of rankings in R which favor x_i over x_j, compared to the number of times x_j is preferred to x_i. Summing a column j of C captures the overall pairwise advantage given to the item x_j in R.

Fairness Threshold. Recognizing that different parity requirements may arise in various ranking scenarios, we now relax our rank parity definition by introducing a fairness threshold parameter δ_{par} as the maximum allowable difference in the pairwise advantage in ρ^* for each group. This supports us in tuning the fairness criterion according to application domain requirements, trading-off between the accuracy of the consensus ranking and the strictness of parity between groups. The threshold is expressed as a proportion of mixed pairs in ρ^*, such that:

$$|Rpar_{G_1}(\rho^*) - Rpar_{G_2}(\rho^*)| \leq \delta_{par} \tag{9.12}$$

9.4.1 Fair-ILP

We now draw on key strategies for tackling the complexity challenge of the aggregation problem, beginning with an Integer Linear Programming (ILP) approach. Exact Kemeny aggregation has been shown to be NP-hard [13, 60], but can be expressed as a variation of the minimum weighted feedback arc set problem [43, 138], and solved using an integer linear program approach [43]. We propose a Fair-ILP by modeling pairwise statistical parity as a constraint in Linear Program 1.

Linear Program 1: Fair-ILP.

$$\text{minimize} \sum_{i,j} C_{ij} A_{ij} \tag{9.13}$$

$$\text{s.t.} \quad A_{ij} \in 1, 0 \tag{9.14}$$

$$A_{ij} + A_{ji} = 1 \qquad \textit{strict ordering constraint} \tag{9.15}$$

$$A_{ij} + A_{jk} + A_{ki} \geq 1 \qquad \textit{transitivity constraint} \tag{9.16}$$

$$\left| \sum_{\substack{x_i \in G_1 \\ x_j \in G_2}} (A_{ij} - A_{ji}) \right| \leq \delta_p \qquad \textit{parity constraint} \tag{9.17}$$

Matrix A specifies the order of all pairs in the desired consensus ranking ρ^*. The first constraint (Equation 9.14) enforces that A is a boolean matrix representing the final ranking ρ^*, wherein each pair appears exactly once. The second constraint (Equation 9.15) says that for all pairs, either $x_i \prec x_j$ or $x_j \prec x_i$, preventing loops in a strict ordering over all candidates. The third constraint (Equation 9.16) enforces transitivity. Conitzer et al. [43] show that together these three constraints are sufficient to produce the Kemeny-optimal aggregation which minimizes the Kendall Tau distance to R.

Equation 9.17 enforces pairwise statistical parity. This constraint sums the differences between entries in A corresponding to mixed pairs in ρ^*, where $A_{i,j}$ represents the pair $(x_i \prec x_j)$ favoring group G_1 over G_2, and $A_{j,i}$ represents its inverse $(x_j \prec x_i)$ favoring group G_2. Summing over these differences computes exactly the difference between the *Rpar* scores as in Equation 9.12.

9.4.1.1 Complexity Analysis for Fair-ILP

While Kemeny aggregation is NP-hard, it is fixed-parameter tractable, depending only on the number of items being ranked n. This intuitively can be understood by considering the size of the n-by-n precedence matrix C. The number of rankings $|R|$ contributes only to a one-time setup cost of $O(|R| * n^2)$ to construct C. The ILP solution for unconstrained Kemeny aggregation requires n^2 binary variables in the matrix A, $\binom{n}{2}$ constraints to enforce strict ordering (Equation 9.15), and $\binom{n}{3}$ constraints for transitivity (Equation 9.16). Our fairness requirement in Equation 9.17 adds additional constraints for each mixed pair ($|G_1||G_2|$ constraints). This, as our experimental study in Section 9.5 confirms, increases the time to solve the program.

For rankings over many items, the large number of binary variables in the ILP solution

poses practical computational challenges. In benchmarking studies [23, 43], ILP algorithms for unconstrained Kemeny aggregation could not handle more than $n = 60$ items. Similarly, in Section 9.5 with state-of-the-art highly optimized mathematical GUROBI solver [76] and 500G of dedicated memory, we handle problems on the order of $n = 100$. Further, with commercial solvers for IP being proprietary and not freely available outside of academia, we explore additional solutions next.

9.4.2 Fair-BB

We now design a Branch-and-Bound based (B+B) approach aiming to handle a larger number of candidates n. Intuitively, any fairness constraints imposed on the rank aggregation task shrink the space of possible outcomes. To capitalize on this problem characteristic, one simple approach would be to incorporate an explicit check into the branching rule for a B+B solution for traditional Kemeny aggregation. This would prune search paths which violate the targeted fairness criterion F. However, we observe that this may require the method to backtrack over many paths in the tree, resulting in an expensive search. Alternatively, to empower more efficient search, we now derive a lower bound on the distance from the base rankings in R to any fair consensus ranking. This can be thought of as the *cost* of each potential solution in terms of pair inversions between rankings. We use this lower bound to design *admissible* fairness-preserving heuristics. which can guide the B+B tree search, and *are guaranteed to underestimate the true cost of the final ranking.* This ensures that whenever a leaf node is reached, an optimal solution has been found [127].

9.4.2.1 Bounding the Cost of Fairness

The total cost of a potential consensus ranking ρ corresponds to the sum of the costs for all ordered pairs of candidates $(x_i \prec x_j)$ in the ranking as given by the Kendall Tau distance (Equation 9.1). This cost for each pair is computed by counting the proportion of rankings in R that disagree with its ordering using the precedence matrix C, such that:

$$cost(x_i \prec x_j) = C_{j,i}. \tag{9.18}$$

Each pair can only be ordered one of two ways, either $(x_i \prec x_j)$ or $(x_j \prec x_i)$. This order will agree with a certain number of rankings in R. Thus, for all $x_i, x_j \in X$, we have:

$$C_{i,j} = 1 - C_{j,i}. \tag{9.19}$$

Summing the minimum costs associated with each pair in the ranking ($C_{i,j}$ or $C_{j,i}$) provides a lower bound $lb(\rho^k)$ on the true cost of the Kemeny optimal consensus ranking [120], as given in Equation 9.20. We note here that this order may contain cycles, and therefore may not correspond to the actual cost of a feasible solution.

$$lb(\rho^k) = \sum_{x_i, x_j \in X} min(C_{j,i}, C_{i,j}) \tag{9.20}$$

Building on this, we now propose Lemma 9.1 following directly from Definition 9.2 of the fair rank aggregation problem.

Lemma 9.1. *Let a Kemeny optimal consensus ranking over X be any ranking ρ^k with minimum cost distance to R, and a fair consensus ranking over X be any ranking ρ^* with minimum cost out of all rankings that satisfy a given fairness criterion F. Then $cost(\rho^*) \geq cost(\rho^k)$.*

Proof. If a ranking ρ^k exists that satisfies F, then $\rho^k = \rho^*$. Otherwise, ρ^* must have higher cost by definition. □

In a higher cost solution, some number $k \geq 0$ pairs must not conform to the minimum cost ordering in the final ranking ρ^*. Assuming we knew this number k, we could then compute a lower bound $lb'(\rho^*)$ based on Equation 9.20 by flipping the number of the k pairs that add the *minimum amount of cost overhead* to $lb(\rho^k)$. Lemma 9.2 identifies the *overhead cost* of flipping a pair.

Lemma 9.2. *Given a pair $(x_i \prec x_j)$ with a minimum cost of $C_{j,i}$ contributing to the sum in $lb(\rho^*)$, if the pair is flipped, the **updated bound** $lb'(\rho^*)$ will incur an **overhead cost** of $1 - (2 * C_{j,i})$.*

Proof. By Equation 9.19, flipping a pair is equivalent to subtracting the minimum cost $C_{j,i}$ for the pair and adding its inverse $(1 - C_{j,i})$. Therefore, when we flip a pair from the order $(x_i \prec x_j)$ to order $(x_j \prec x_i)$, we get:

$$\begin{aligned} lb'(\rho^k) &= lb(\rho^k) - C_{j,i} + (1 - C_{j,i}) \\ &= lb(\rho^k) + 1 - 2 * C_{j,i} \end{aligned}$$

□

Corollary 9.1. *$lb'(\rho^*)$ can be determined from the ordering of pairs used to compute $lb(\rho^k)$ by flipping the k pairs that carry the minimum overhead costs, and that will satisfy the criterion F.* ■

Proof. Corollary 9.1 follows directly from Lemmas 9.1 and 9.2, given that $C_{i,j} \geq C_{k,l} \implies 1 - 2 * C_{i,j} \leq 1 - 2 * C_{k,l}$ □

9.4.2.2 Guiding Search for Fair Consensus Ranking

Building on the observations above, we now incrementally construct a search tree such that each node v represents a candidate x being placed in a particular rank position in ρ^*. Each node is expanded by adding children nodes for all items not yet included in the path to v. The full search tree has depth n, with every path through the tree representing one possible ranking, and $n!$ leaf nodes. To bound the search and avoid traversing the entire tree, as each node is expanded, we compute a two-part cost function $f(v) = g(v) + h(v)$.

We can think of $g(v)$ as the cost of the *pairs-so-far* set by the order of the candidates in the prefix path from the root to the current node v. Heuristic $h(v)$ models an estimate of the cost for the *pairs-to-go* which are yet to be determined. Given a node v, the *pairs-so-far* set by the path to v can be divided into three subsets: pairs of candidates belonging to the same group, pairs favoring group G_1 over G_2 (denoted $v.p1$), and pairs favoring group G_2 over G_1 (denoted $v.p2$). These pairs all contribute to the cost of the *pairs-so-far* $g(v)$.

Rank Parity Heuristic. To determine $h(v)$ for rank parity, we initially assign all *pairs-to-go* their minimum cost ordering as required to compute $lb(\rho^k)$ (Equation 9.20). Once all the pairs are ordered, they can similarly be divided into subsets. We denote *pairs-to-go* favoring G_1 over G_2 as $v.p2g1$, and pairs favoring G_2 over G_1 as $v.p2g2$. We can now express the rank parity criterion in Equation 9.12 for a node v as the following requirement:

$$abs(\ |v.p1 \cup v.p2g1| - |v.p2 \cup v.p2g2|\ ) \leq \delta_{par} \tag{9.21}$$

where $|\cdot|$ denotes the number of pairs in each set. If this condition is not met, we must update our initial ordering of *pairs-to-go* to be sure it only allows for rankings which satisfy rank parity. This is accomplished by flipping the order of some of the *pairs-to-go*, transferring them from the set favoring one group to the set favoring the other and balancing the pairwise advantage of the groups. We determine the required number of pairs to flip to satisfy rank parity k, according to Lemma 9.3.

Lemma 9.3. *Given a node v, let $P_{max} = max(|v.p1 \cup v.p2g1|, |v.p2 \cup v.p2g2|)$ and $P_{min} = min(|v.p1 \cup v.p2g1|, |v.p2 \cup v.p2g2|)$. **The number of pairs to flip** to satisfy Equation 9.21 is:*

$$k = \left\lceil \frac{P_{max} - P_{min} - \delta_{par}}{2} \right\rceil$$

Algorithm 1: getParityOverhead

input : node v
output: overhead cost for v

if $|v.p_1 + v.p2g_1| > |v.p_0 + v.p2g_0|$ **then**
  $k = \left\lceil \frac{|v.p_1+v.p2g_1|-\delta}{2} \right\rceil$;
  pairsList $\longleftarrow v.p2g_1$;
else if $|v.p_1 + v.p2g_1| < |v.p_0 + v.p2g_0|$ **then**
  $k = \left\lceil \frac{|v.p_0+v.p2g_0|-\delta}{2} \right\rceil$;
  pairsList $\longleftarrow v.p2g_0$;
else
  $k = 0$
if $k == 0$ **then**
  return 0
else if $k \leq$ `length(pairsList)` **then**
  `sort(pairsList)` ; `// sort in ascending order of overhead cost`
  overhead $\longleftarrow 0$;
  for i **to** k **do**
    overhead $+= 1 - 2 *$ `cost(pairsList[`i`])`;
else
  overhead $\longleftarrow \infty$; `// constraint cannot be met`
return overhead

Proof. When we flip a pair, we subtract 1 from P_{max} and add it to P_{min}. Therefore we simply can derive the number of pairs to flip as:

$$\begin{aligned}(P_{max} - k) - (P_{min} + k) &= \delta_{par} \\ P_{max} - P_{min} - 2k &= \delta_{par} \\ \left\lceil \frac{P_{max} - P_{min} - \delta_{par}}{2} \right\rceil &= k \quad \square\end{aligned}$$

Given the required number of pairs to flip k, we can then check whether there are a sufficient number of *pairs-to-go* that could be flipped. If not, this path cannot yield a feasible solution and can thus be pruned. If there are more than enough *pairs-to-go*, then following Corollary 9.1, flipping only those k pairs that add the minimum overhead will give us our adjusted lower bound on the cost of *pairs-to-go* $lb'(\rho^*)$. Algorithm 1 gives the procedure for computing the total minimum overhead cost for k flipped *pairs-to-go*.

9.4.2.3 Complexity Analysis for B&B Solution

Complexity of Search Tree Exploration. In the worst case, every path in the tree will have to be explored yielding an $O(n!)$ search cost. Even for modest values of n, such a search will likely be intractable. However, in [120], Meila et al. observe that performance is greatly impacted by the amount of agreement among the base rankings in R. If there is little agreement, all rankings in S_n will be far from the set R and many paths in the tree will have to be compared. In contrast, if there is strong agreement in R, then only a small number of rankings in S_n will be likely candidates for ρ^*. Given a good search heuristic, only the small number of likely rankings will need to be explored.

The cost of the B+B algorithm also depends on the search strategy used. We implement A* search, using a priority queue which has constant time cost for adding and removing nodes if a Fibonacci heap isused. Overall performance of the search depends heavily on the heuristic used. In Section 9.5 we empirically evaluate our rank parity heuristic given varying degrees of agreement among the rankings.

Complexity to Compute Parity Heuristic. As is typical in B+B design, there is a complexity tradeoff between computing a tighter lower bound heuristic and its impact on the resulting search [40]. Each time a node v is expanded, we must compute the cost of the *pairs-so-far* $g(v)$ and cost of the *pairs-to-go* $h(v)$ for every child node. Following the strategy in [120], we use the siblings of each node to compute $g(v)$, and to determine the sets of *pairs-to-go*. This requires constant time complexity and $O(n)$ space complexity to store the child nodes. To compute the parity heuristic $h(v)$, Algorithm 1 sorts the *pairs-to-go* in P_{max}. Sorting all $n(n-1)/2$ candidate pairs has complexity $O(n^2(log(n))$. Therefore, traversing a single path from the root to a leaf, we visit n nodes and expand $O(n)$ child nodes. This results in $O(n^4log(n))$ time complexity.

9.4.3 Fair-Post

To handle the case when the B+B method exceeds allowable resources, one could revert to an approximate tree search using standard techniques such as best first search or beam search. Unfortunately, this simple approach guarantees neither optimal distance to the base rankings nor fairness. As an alternative, we now return to the post-processing strategy discussed in Section **??** to design a approximation strategy which not only guarantees pairwise statistical parity, but also assures that a minimum number of pair inversions are introduced. Given this, existing fast approximation methods for Kemeny aggregation can be plugged in to first aggregate R and generate an initial consensus

ranking, and then correct for fairness while bounding the additional approximation error added.

Our Fair-Post is given in Algorithm 9.4a. The input is an *unfair* ranking $\rho = [x_1 \prec \cdots \prec x_n]$ which does not satisfy pairwise statistical parity. Let us say without loss of generality that more pairs favoring group G_1 is greater than the number of pairs G_2. A fair version of this ranking should only allow a certain number of pairs to favor G_1 – denoted $maxPairs$. To correct the ranking, the algorithm proceeds by forming two queues of candidates l_1, l_2 for each group, respectively, keeping the candidates in their original rank order. Then, iterating through rank position $i = 1$ to n, at each step the candidate ranked highest in ρ in either queue is placed at the current position i in ρ', provided that the fairness constraint will not be violated. To check this condition, we keep a tally of the *pairs-so-far* favoring each group denoted p_1 and p_2, respectively. If the selection of a candidate from G_1 would cause the number of pairs favoring G_1 to exceed $maxPairs$ then parity would be violated (line 11). In this case the highest ranked item from queue l_2 is chosen instead.

Number of *pairs-so-far*. Each time a candidate x is placed at position i, it forms pairs of items favoring x over all subsequent candidates. If the candidate is in group G_1, the number of mixed pairs created is the number of remaining candidates in l_2 yet to be ranked. The count of *pairs-so-far* p_1 is updated accordingly (line 13). When a candidate from G_2 is chosen, p_2 is updated in a similar fashion based on the number of items in l_1 (line 21).

Number of *pairs-to-go*. At each step i, the mixed pairs left to be formed are between the candidates in the queues l_1 and l_2. Therefore the number of mixed *pairs-to-go* is $|l_1^i||l_2^i|$.

Total number of mixed pairs. Following from the observations above, at any step i, the total number of mixed pairs m is made up of the *pairs-so-far* and *pairs-to-go*. Therefore:

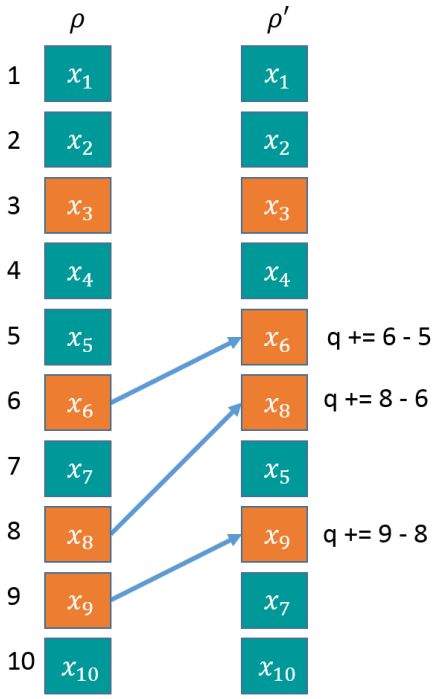
$$m = p_1^i + p_2^i + q^i + |l_1^i||l_2^i| \tag{9.22}$$

Number of flipped pairs. In the process of correction, some number q pairs will be flipped. Pairs are only ever flipped from favoring $G_1 \prec G_2$ in the original ranking ρ to now favor $G_2 \prec G_1$ in the output ranking ρ'. Choosing a lower ranked item from G_2 to satisfy parity creates mixed pairs favoring $x \in l_2$ over all the candidates in l_1. Many of those pairs have the same order as in the original ranking, however some will

: correctParity

input : $\rho = [x_1 \prec x_2 \prec \cdots \prec x_n]$, maxPairs
output: ρ' corrected ranking, $q = K(\rho, \rho')$

$l_1, l_2 \longleftarrow$ empty queues;
for $i \leftarrow 1$ **to** n **do**
  if $x_i \in G_1$ **then**
    insert x_i into l_1;
  else
    insert x_i into l_2;
$\rho' \longleftarrow [\,]$;
$p_1, p_2, q \longleftarrow 0$;
for $i \leftarrow 1$ **to** n **do**
  if $\rho(l_1.\texttt{peek()}) > \rho(l_2.\texttt{peek()})$ **then**
    if $p_1 + l_2.\texttt{length()} \leq$ maxPairs **then**
      $\rho'[i] = l_1.\texttt{dequeue()}$;
      $p_1 = p_1 + l_2.\texttt{length()}$;
    else
      $flip = \rho(l_2.\texttt{peek()}) - i)$;
      $\rho'[i] = l_2.\texttt{dequeue()}$;
      $p_2 = p_2 + l_1.\texttt{length()} - flip$;
      $q = q + flip$;
  else
    $\rho'[i] = l_2.\texttt{dequeue()}$;
    $p_2 = p_2 + l_1.\texttt{length()}$;
return ρ', q



be flipped. The number of flipped pairs is $\rho(x) - i$. We can see this in the figure above when candidate x_8 is placed at position $i = 6$ resulting in two flipped pairs ($x_8 \prec x_5$) and ($x_8 \prec x_7$). We track the number of flipped *pairs-so-far* q separately from the number of pairs favoring G_2 retaining the same order as ρ (lines 15-18).

9.4.3.1 Proof of Fairness and Optimal Aggregation Error

To prove that the correctParity Algorithm produces an optimal fair result, we first prove that if the output ranking ρ' satisfies the pairwise statistical parity criterion, then a minimal number of pair inversions will have been made, resulting in an optimal Kendall Tau distance between ρ and ρ'. Then we prove by induction that the algorithm guarantees

a fair result. Relevant notation is defined in Table 9.1.

Symbol	Definition
ρ	Input unfair ranking.
ρ'	Corrected output ranking.
$\rho(x)$	Rank of candidate x in ρ.
l_1^i, l_2^i	Queues of candidates from each group G_1, G_2 in order of rank in ρ at step i.
h	The minimum number of pairs to flip.
q^i	The number of flipped pairs so far at step i.
p_1^ρ, p_2^ρ	Number of pairs favoring group G_1, G_2 in original ranking ρ.
p_1^i, p_2^i	Number of *pairs-so-far* favoring group G_1, G_2 at step i.

Table 9.1: Table of symbols.

Let p_1^ρ and p_2^ρ denote the number of pairs favoring groups G_1 and G_2 respectively in the original ranking ρ, such that $p_1^\rho + p_2^\rho = m$. There exists some **minimum number of pairs** $h > 0$ that must be flipped in order for ρ' to satisfy rank parity, such that:

$$p_1^\rho - h = maxPairs = m - (p_2^\rho + h) \tag{9.23}$$

At each step i of the algorithm we have:

$$p_1^i \leq p_1^\rho, \text{ and } p_2^i \leq p_2^\rho \quad \forall\ 0 < i \leq n \tag{9.24}$$

At the last step n of the algorithm, $p_2^n = p_2^\rho$, and q^n is the total number of pair inversions between ρ and ρ', corresponding to the Kendall Tau distance between the original ranking and the corrected version. If $p_1^n = maxPairs$ and $p_2^n + q^n = m - maxPairs$ then the output ranking ρ' satisfies parity. If $q^n = h$ then we have flipped only the minimum number of pairs, and the Kendall Tau distance $K(\rho, \rho')$ is optimal.

Proof that a fair result is optimal. We now show that for all steps of the algorithm:

$$q^i \leq h \iff p_2^i + q^i \leq m - maxPairs \tag{9.25}$$

First we show that for all i, $q^i \leq h \implies p_2^i + q^i \leq m - maxPairs$:

$$
\begin{aligned}
q^i &\leq h \\
p_2^i + q^i &\leq p_2^i + h \\
p_2^i + q^i &\leq p_2^\rho + h && \text{by Equation 9.24} \\
p_2^i + q^i &\leq m - maxPairs && \text{by Equation 9.23}
\end{aligned}
$$

Next we show that $p_2^i + q^i \leq m - maxPairs \implies q^i \leq h$ using proof by contraposition. Suppose that $q^i > h$. We have:

$$
\begin{aligned}
m &> p_1^i + p_2^i + h + |l_1^i||l_2^i| && \text{by Equation 9.22} \\
m - h &> p_1^i + p_2^i + |l_1^i||l_2^i| \\
p_2^\rho &> p_1^i + p_2^i + |l_1^i||l_2^i| && \text{by Equation 9.23} \\
p_2^\rho &> m - q^i && \text{by Equation 9.22} \\
q^i &> m - p_2^\rho = p_1^\rho && \text{by definition of } p_1^\rho \\
q^i &> maxPairs && \text{by Equation 9.23} \\
p_2^i + q^i &> maxPairs \geq m - maxPairs && \text{by definition of } maxPairs
\end{aligned}
$$

Having proved the contrapositive $q^i > h \implies p_2^i + q^i > m - maxPairs$, we can infer that Equation 9.25 is true for all positions in the ranking ρ'. □

Proof of fairness. We now prove by induction that the correctParity algorithm produces a ranking ρ' which satisfies statistical parity by showing that for all rank positions i, Equations 9.26 and 9.27 hold. We consider only cases where $n \geq 3$ in which there are more than one pair of candidates.

$$p_1^i \leq maxPairs \qquad (9.26) \qquad p_2^i + q^i \leq m - maxPairs \qquad (9.27)$$

Base case. Prior to adding any candidates to ρ', all values p_1^0, p_2^0, q^0 are equal to 0. At step 1, there are 3 ways we can choose the first candidate in the ranking:

1. *The highest ranked candidate in ρ is $x \in l_2$.*

We have $p_1^1 = p_1^0 = 0$ and $q^1 = q^0 = 0$. To prove Equation 9.27 we observe:

$$\begin{aligned} p_2^1 + q^1 &\leq p_2^\rho + 0 && \text{by Equation 9.24} \\ p_2^\rho + 0 &\leq m - maxPairs && \text{by Equation 9.23} \end{aligned}$$

2. *The highest ranked candidate in ρ is $x \in l_1$, and choosing it does not violate parity.*

   Since the parity condition holds, $p_1^1 \leq maxPairs$. We form no flipped pairs so $q^1 = 0$ and $p_2^1 + q^1 = p_2^0 + q^0 = 0 < m - maxPairs$.

3. *The highest ranked candidate in ρ is $x \in l_1$ but choosing it would violate the parity constraint, so we choose a lower ranked candidate $y \in l_2$.*

   This only happens if $p_1^0 + |l_2| > maxPairs$. We know that $maxPairs \geq \frac{m}{2}$ by definition, and since all items are still in the queues, $m = |l_1^1||l_2^1|$. Therefore:

$$|l_2^1| > \frac{m}{2} \implies |l_2^1| > \frac{|l_1^1||l_2^1|}{2} \implies |l_1^1| = 1$$

   This means that there is only one candidate in G_1, originally ranked in the top spot. We choose a candidate from G_2 instead, forming 1 flipped pair, so $q^1 = 1$ and $p_2^1 = p_2^0 = 0$. We know $n = |l_1^1| + |l_2^!| \geq 3 \implies m \geq 2$. Therefore $p_2^1 + q^1 = 1 \leq m - maxPairs$.

Induction step. Suppose that at step i, $p_1^i \leq maxPairs$, and $p_2^i + q^i \leq m - maxPairs$. At step $i+1$, there are again 3 ways of choosing a candidate:

1. *The next highest ranked candidate is $x \in l_2$.*

   We have $p_1^{i+1} = p_1^i \leq maxPairs$ by induction hypothesis. Since x is the highest ranked candidate we don't form any flipped pairs, so $q^{i+1} = q^i$. Therefore:

$$\begin{aligned} p_2^{i+1} + q^{i+1} &= p_2^{i+1} + q^i \\ p_2^{i+1} + q^{i+1} &\leq p_2^\rho + q^i && \text{by Equation 9.24} \\ p_2^{i+1} + q^{i+1} &\leq p_2^\rho + h && \text{by Equation 9.25} \\ p_2^{i+1} + q^{i+1} &\leq m - maxPairs && \text{by Equation 9.23} \end{aligned}$$

2. *The next highest ranked candidate is $x \in l_1$, and choosing it does not violate parity.*

Since the parity condition is not violated, $p_1^{i+1} \leq maxPairs$. No pairs favoring G_2 are formed, so $p_2^{i+1} + q^{i+1} = p_2^i + q^i \leq m - maxPairs$ by induction hypothesis.

3. *The next highest ranked candidate is $x \in l_1$, but choosing it would violate the parity constraint, so we choose a lower ranked candidate $y \in l_2$.*

   We know that $p_1^{i+1} = p_1^i \leq maxPairs$ by induction hypothesis.

   To show that $p_2^{i+1} + q^{i+1} \leq m - maxPairs$ we give a proof by contradiction. Suppose $p_2^{i+1} + q^{i+1} > m - maxPairs$. Since parity would have been violated if we chose $x \in |l_2^i|$, we also have $p_1^i + |l_2^i| > maxPairs$. From this we can derive:

$$\begin{aligned} p_2^{i+1} + q^{i+1} + p_1^i + |l_2^i| &> m \\ p_2^i + |l_1^i| - (\rho(y) - 1) + q^i + (\rho(y) - 1) + p_1^i + |l_2^i| &> m \\ |l_1^i| + |l_2^i| &> m - (p_1^i + p_2^i + q^i) \end{aligned} \tag{9.28}$$

   At step i, $(p_1^i + p_2^i + q^i)$ is simply the total number of mixed *pairs-so-far* in the ranking. Since the *pairs-so-far* and *pairs-to-go* make up the total number of mixed pairs m (Equation 9.22), we have $m - (p_1^i + p_2^i + q^i) = |l_1^i||l_2^i|$. Therefore Equation 9.28 can be reduced to $|l_1^i| + |l_2^i| = |l_1^i||l_2^i|$. However, the size of the queues are strictly positive integers, therefore their sum cannot be greater than their product. That is $a + b > ab \implies a > ab - b \implies ab - b + b > ab \implies ab > ab$ which cannot be true.

Conclusion. By the principle of induction, Equations 9.26 and 9.27 are true for all positions i in the ranking ρ'. □

9.4.3.2 Complexity Analysis for Fair-Post

Algorithm 9.4a has O(n) time complexity, requiring two passes over the input ranking first to populate the queues and then to assign the candidates to their corrected rank positions. It has O(n) space complexity to hold the candidates in the queues. Given we first need to aggregate the rankings which may contain a large number of items, we propose to plug in an existing approximate aggregation method as an initial step. This way we address the complexity concern of the alternate exact aggregation solutions such as the NP-hard Kemeny aggregation, and the (non-fair) ILP [43] and B+B [120] solutions - with the later having similar limitations on the number of candidates to be ranked as our proposed integrated methods.

9.5 Evaluation

9.5.1 Experimental Methodology

Overall Strategy. We conduct a systematic study analyzing the interplay between critical factors including # candidates being ranked, # rankings in R, consensus among rankings in R, and group fairness threshold on the fair rank aggregation algorithms. Then we conduct a case study using real-world sports ranking data generated by human decision makers evaluating other people.

Metrics. To measure the accuracy of the aggregation, we use the average Kendall Tau distance to the rankings in R (Equation 2), denoted $K_{mean}(\rho^*)$. To evaluate the fairness (*pairwise statistical parity*) achieved by the consensus ranking ρ^*, we measure the absolute difference in the $Rpar_{G_i}$ scores (Definition 9.4) for each group, denoted as $Rpar(\rho^*)$ in Equation 9.29. Since the $Rpar_{G_i}$ scores for each group are normalized by the total number of mixed pairs, the *Rpar* score is equal to 1 when the ranking is totally biased favoring one group, and 0 when each group is favored in half of the pairs for perfect parity.

$$Rpar(\rho^*) = abs(Rpar_{G_1}(\rho^*) - Req_{G_2}(\rho^*)) \tag{9.29}$$

Methods. As a baseline, we compare against two existing strategies for (nonfair) exact Kemeny aggregation: integer linear program (ILP) by Conitzer et al. [43], and a B+B algorithm by Mandhani and Meila [117] which uses the lower bound in Equation 9.20. When evaluating approximate aggregation, we compare against the classic Borda [50] scoring method. In a comparative study of algorithms for Kemeny aggregation [5], Borda was suggested as the best approximation approach when optimizing for speed, and to give low approximation error. Borda simply sorts the candidates by the the overall pairwise advantage in R, which can be done efficiently by summing the columns of the precedence matrix. We denote our proposed fairness-preserving exact methods as Fair-ILP for the integer-linear-program with rank parity constraint (Section 9.4.1) and Fair-BB for our B&B method with rank parity-preserving heuristic (Section 9.4.2). We apply our Fair-Post algorithm introduced in Section 9.4.3 as post-processing step to the unconstrained exact and approximate methods.

Experimental Setup. All experiments were performed on a Linux server running Ubuntu 14 with 500G of RAM. Integer programming solutions were implemented using the commercial highly optimized and parallelized mathematical solver GUROBI [76]. Borda and Fair-Post methods we implemented in Python. B+B algorithms were imple-

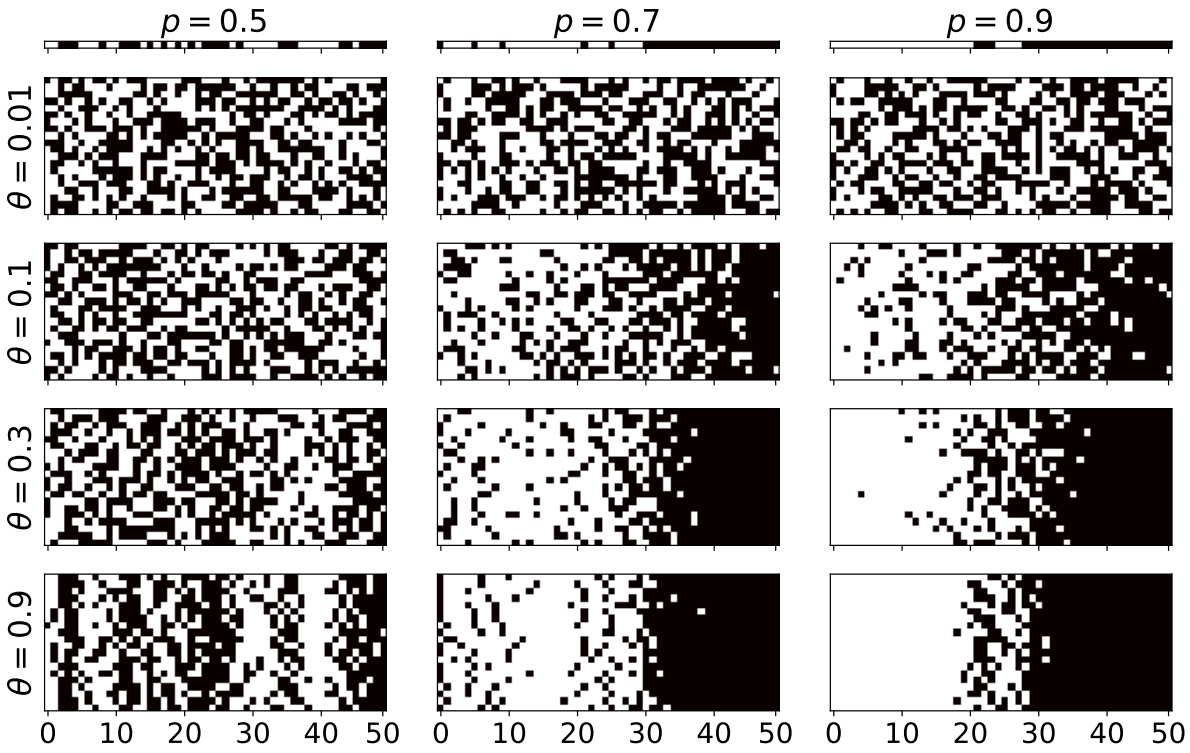


Figure 9.5: Impact of parameters θ controlling consensus, and p controlling fairness on sets of Mallows generated base rankings with $n = 50$ items and $|R|$=20.

mented in Java, adapted from implementation by authors of [117]. For Java methods, we fix the heap size of JVM at 50G. Direct comparisons of run-times should consider these differences.

9.5.2 Controlled Study of Fair Kemeny Aggregation using Mallows Model

Dataset Generation. We adopt the Mallows Model probability distribution over rankings [116] which provides a natural means to evaluate Kemeny rank aggregation methods extensively in previous studies [5, 23]. The Kemeny optimal consensus ranking has been shown to be a maximum likelihood estimator for this model [157]. For all rankings

$\pi \in S_n$, the Mallows model is the probability distribution:

$$P_{\pi_0,\theta}\frac{exp(-\theta K(\pi,\pi_0))}{Z}$$
$$Z = \prod_{i=1}^{n-1}\frac{1-exp((-n-i+1)\ \theta_i)}{1-exp(-\theta_i)} \tag{9.30}$$

The distribution is parameterized by θ which controls the degree of consensus among the rankings in R. If $\theta = 0$, Equation 9.30 yields a uniform distribution, i.e., there is no consensus among the rankings. As θ increases, the distribution becomes steeper around a single mode ranking σ_0.

To understand the fairness of ρ^* compared the fairness of the base set of rankings in R, we introduce a second parameter p. We control parity in the base rankings by assigning the candidates in the central ranking σ_0 to two groups G_1 and G_2, starting from the highest ranked item and progressing sequentially to the lowest ranked. For each candidate, the group is chosen with probability p. For $p = 0.5$, the groups are assigned in a uniform random manner. Given enough items, this central ranking will be fair, i.e., $Rpar(\sigma_0)$ will be close to 0. As p increases, group G_1 is more likely to be chosen, leading to more candidates from G_1 appearing in favorable positions in the ranking. When $p = 1$, all candidates in G_1 are ranked above those in G_0, resulting in a completely biased ranking with $Rpar(\sigma_0) = 1$.

9.5.2.1 Descriptive Study of Consensus and Fairness in Mallows Data

Figure 9.5 shows twelve different sets of base rankings generated using the Mallows model with $n = 50$ candidates and $|R| = 20$ rankings. We create different aggregation scenarios by varying the parameters θ and p. Items from group G_1 are colored white, and items from G_0 are black. The top row shows the central rankings σ_0 used to generate four versions of R in the corresponding columns, with highly ranked items on the right. Moving top to bottom, θ is first close to 0, producing little consensus among the rankings. As θ is increased moving down, the rankings in R tend to agree more and more with the order of items in σ_0.

In each column of Figure 9.5 the candidates are assigned to groups with different degrees of bias. On the left, items are assigned with probability $p = 0.5$, and both groups are randomly distributed throughout all sets of rankings, even when there is a high degree of consensus. In the middle, σ_0 is generated with $p = 0.7$, and on the right

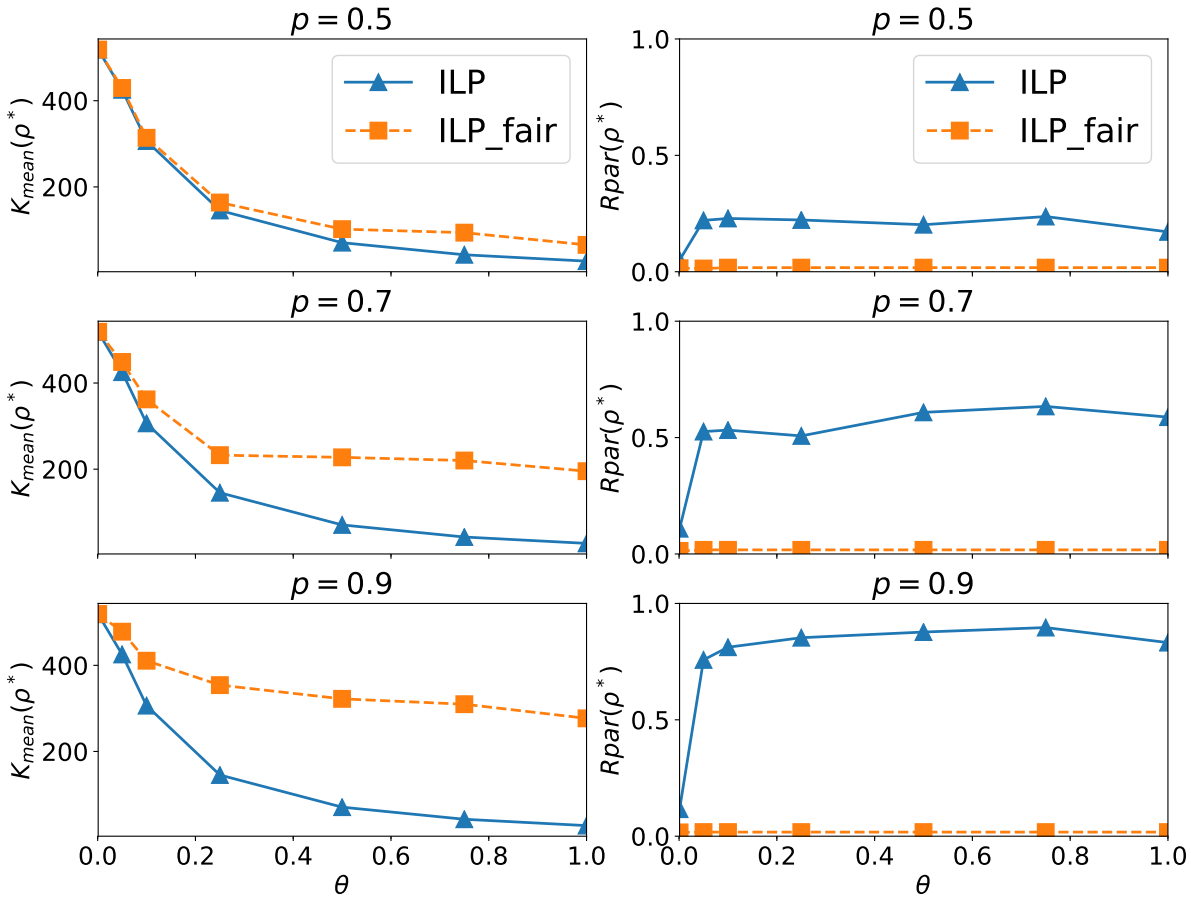


Figure 9.6: Impact of agreement in R on distance to ρ^* (left) and on the rank parity of ρ^* (right) on sets of Mallows generated base rankings with $n = 50$ items and $|R|$=20 for unconstrained and fairness-preserving ILP methods.

$p = 0.9$. Group G_1 is increasingly favored over group G_0. We observe that when θ is small, all the datasets are noisy, no matter the value of p. As θ increases and the rankings begin to agree, a distinct advantage is introduced for group G_1 in R corresponding to the unfairness controlled by p (Figure 9.5, bottom right).

This demonstrates that consensus among rankings in R has a large impact on the overall fairness. When there is little consensus (top row of Figure 9.5), any bias in an individual ranking is "cancelled out" by the diversity in base rankings. In this case, we expect to see little penalty in aggregation accuracy when enforcing fairness in ρ^*. Any ranking chosen will have a large distance to R given rankings are so dissimilar from each other. Conversely, when θ is large (bottom row Figure 9.5), the distance to the consensus ranking will be small, and accuracy tradeoff for fairness will be higher.

9.5.2.2 Experimental Study using Mallows Data

Next we demonstrate that our fair aggregation methods learn accurate consensus rankings, while enforcing fairness - even when base rankings are unfairly biased.

Accuracy versus Fairness Tradeoff. First, we verify observations in our descriptive study using unconstrained ILP and parity-preserving Fair-ILP in the same setting. To reveal the impact of strict fairness criteria on aggregation accuracy, we set a tight fairness threshold δ_{par} allowing an advantage of at most 0.02% of the mixed pairs for either group. On left, Figure 9.6 shows the impact of enforcing parity in ρ^* on the average Kendall Tau distance to R for base rankings with different degrees of agreement and different amount of bias. When θ is close to zero, R is very noisy, and therefore the overall $dist(\rho^*)$ is high, for both unconstrained aggregation or fair consensus ranking. As θ increases, there is more agreement among rankings in R, and the distance to ρ^k found by ILP gets smaller. Fair-ILP carries a higher distance penalty for requiring parity in ρ^*, which becomes more pronounced with stronger unfair bias in R ($p = 0.9$).

Relationship between Rank Parity and Consensus. Figure 9.6 (right) shows $Rpar(\rho^*)$ scores (Equation 9.29) on y-axis. The pairwise statistical parity of the ρ^k ILP solution reflects the unfair advantage in base rankings introduced by parameter p. Fair-ILP succeeds in enforcing the parity threshold, yielding a flat score across all values of θ and p.

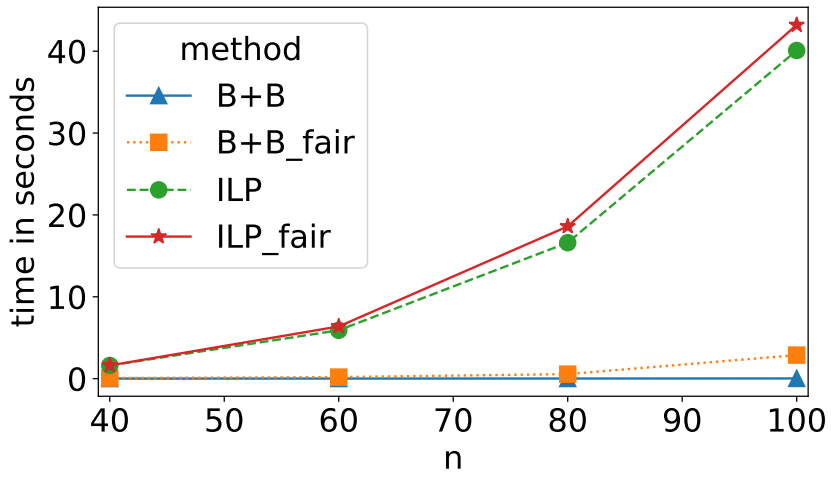


Figure 9.7: Comparison runtimes for unconstrained and fairness-preserving B+B and ILP methods on sets of $|R| = 1000$ Mallows generated base rankings with $theta = 0.3$, $p = 0.7$ and fairness threshold of 25% mixed pairs.

9.5.2.3 Performance Evaluation

Next we compare the performance of our proposed exact methods Fair-ILP and Fair-BB for fair rank aggregation. For this, we generate datasets using the Mallows model varying the number of candidates from $n = 40$ to 100. We did not observe significant impact on run-times due to the number of rankings being aggregated. Therefore, we fix $|R| = 1000$.

Impact of number of candidates on performance. From our cost analysis in Section 9.4.1.1, we know that the number of candidates being ranked is the most important determinant of time complexity of the Fair-ILP methods. Indeed, the Fair-ILP method proves to be robust across parameter settings for θ and different fairness thresholds, achieving consistent run-times across all settings. The parity constraint introduced in Section 9.4.1 adds some overhead impacted by the amount of bias in the base rankings. However, as Figure 9.7 shows, the ILP run-times increase exponentially in the number of candidates n. This confirms similar analysis in previous studies (see 2015 VLDB survey [23] for $n > 60$ candidates).

The B+B approaches tell a different story. In Figure 9.7, we fixed $\theta = 0.3$ and generated biased data with $p = 0.7$. A loose fairness threshold allowed for an advantage of 25% of the mixed pairs. In this case, Fair-BB gives much better performance than Fair-ILP, scaling linearly in the number of candidates. However, the B+B approach is much more sensitive to the amount of bias in R and agreement among the rankings. This is expected, since the worst case complexity of the method is $O(n!)$ as discussed in Section 9.4.2.3.

Impact of Unfair Bias on Fair-BB. For their unconstrained B+B method for Kemeny aggregation, [120] claims that the worst case is avoided when the base rankings strongly agree. We observe that for fair aggregation, too much agreement can hinder performance. When strict fairness is required and biased base rankings agree strongly, Fair-BB performance suffers greatly. In our experiments, a number of parameter settings required more memory than available with heap size of 50GB. It may be that when θ is large, the base rankings are tightly clustered far from all potentially fair solutions. Therefore none can be pruned even using the parity heuristic.

9.5.2.4 Approximation Methods

As discussed in Section 9.3, alternative approaches to our integrated solutions include pre-processing and post-processing strategies using an existing aggregation method along with a fairness correction method to adjust individual rankings to meet the desired fair-

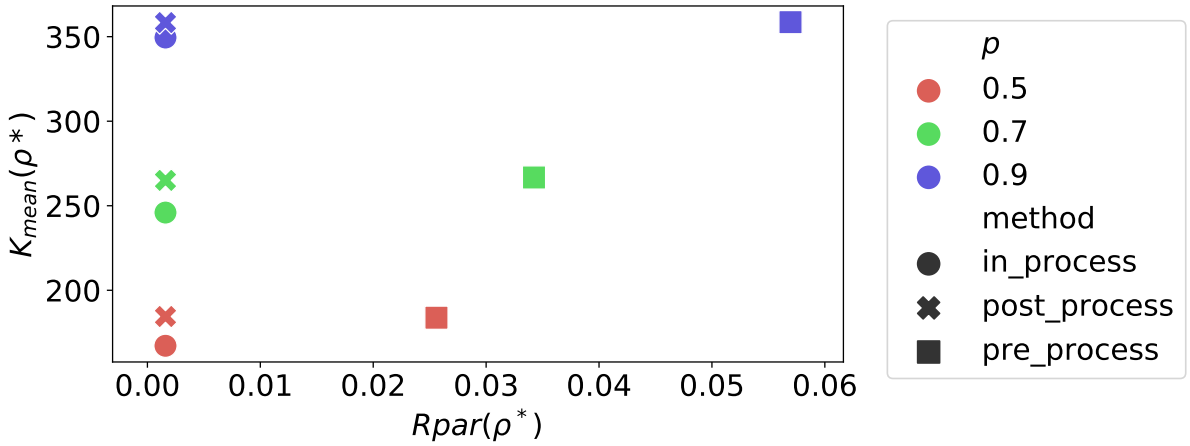


Figure 9.8: Comparison of pre-processing, post-processing, and in-processing fair rank aggregation methods.

ness criteria. We evaluate these alternatives in Figure 9.8 compared to our proposed Fair-ILP method, which guarentees both fairness and optimal aggregation. The methods are implemented using our proposed Fair-Post Algorithm in conjunction with the unconstrained ILP by Conitzer et al. [43] for aggregation.

Input sets of rankings with $|R| = 20$ and $n = 50$ candidates are generated using the Mallows model with θ fixed as 0.3, and different levels of bias $p = 0.5, p = 0.7$ and $p = 0.9$ (indicated by color). We compare the approximation error in terms of average Kendall Tau distance between the consensus ranking and R (on the y-axis) and the pairwise statistical parity of the result (on the x-axis).

The pre-processing strategy, indicated by the square markers, corrects each ranking for fairness and then aggregates the results. As Figure 9.8 shows, the resulting aggregations are neither fair, with *Rpar* scores ranging from 0.02 to 0.06, nor optimal in terms of distance to the base rankings. For post-processing, indicated by x markers, we first run unconstrained aggregation then correct the consensus ranking. In this case the results are fair, with *Rpar* scores close to 0, however they still introduce some aggregation error in terms of distance. Only the in-processing strategy achieves both goals simultaneously.

Scalability of Post-processing Strategy. For databases of many candidates, if we accept an approximate solution our Fair-Post method can easily scale to the handle rankings over thousands to millions of candidates. Table 9.2 gives the runtimes when aggregating rankings of $n = 100$ to $n = 1,000,000$ candidates. We fix $|R| = 1000, p = 0.7$, and fairness threshold of 1% of the mixed pairs as these settings were not observed to impact runtime in our experimental analysis. We see that the Fair-Post used with the

	Time in Seconds	
#Candidates	**Borda-Agg.**	**Fair-Post**
100	0.11	0.30
1,000	0.89	2.41
10,000	8.83	23.48
100,000	104.85	235.07
1,000,000	***	2798 .57

Table 9.2: Impact of number of candidates on run time for post-processing approximate aggregation of $|R| = 1000$.

efficient Borda approximation method scales linearly to easily handle large datasets. For candidates with $n > 100,000$ candidates we report only Fair-Post times for the central rankings σ_0 due to the overhead of generating Mallows data at that scale.

9.5.3 FantasyPros Ranking Case Study

We evaluate our fair aggregation methods using weekly sports rankings from the popular website FantasyPros.[1]

These publicly available rankings of NFL players are created by expert sports analysts to help inform fantasy football picks. This data provides an ideal test-bed reflecting a real-world phenomena where human voters rank and judge other people as candidates. Due to the sensitive nature of protected data attributes and biased decision making, we model unfairness by adding a synthetic protected data attribute to the candidates being ranked (as has been common in other fair ranking literature [70, 155, 161]).

We pulled rankings of wide receivers for the second week of the 2019 NFL season. This position had the largest pool of players to be ranked. We considered only complete rankings over the 50 players ranked by most experts, resulting in a set of 23 expert rankings of 50 players to be aggregated. We first ran an unconstrained ILP Kemeny aggregation to find a baseline consensus ranking ρ^k. Considering this as our central ranking, we then simulate group bias by assigning the candidates to groups with different degrees of fairness. We then aggregate the now biased set of rankings using our Fair-ILP method. We fix a tight fairness threshold of 3 pairs (1% of the mixed pairs). Results are averaged over 10 runs. The average runtime for Fair-ILP was 9 seconds to create fair aggregations.

[1] https://www.fantasypros.com/nfl/rankings

Table 9.3 evaluates the accuracy versus fairness tradeoff for base sets or rankings with varying degrees of bias. The top row gives the average *Rpar* score for unconstrained aggregation, and bottom rows the average added number of pair inversions resulting from fair aggregation. This demonstrates that our methods can easily correct for bias in real-world scenarios. Enforcing rank parity requires an increase of 0.18% average Kendall Tau distance to the rankings in R for the most stringent fairness threshold.

$Rpar(\sigma_0)$	0.0	0.2	0.4	0.6	0.8
$+K\_mean$	0.49	21.93	53.53	66.93	67.32

Table 9.3: Accuracy verses fairness tradeoff for sports ranking data with $|R| = 24$ rankings and $n = 50$ candidates.

10

Related Work

10.1 Evaluating Fairness in Ranking

Top-k Statistical Parity. The majority of early work on fair ranking has focused on statistical parity [34, 70, 155, 161]. This evaluation strategy for fairness was first proposed and considered for classification by Dwork *et al.* [58]. While popular, it has been observed that enforcing statistical parity may exact a high toll in terms of predictive accuracy, and possibly infringe on fairness for individuals [58, 80]. For ranking, a number of evaluation metrics for statistical parity have been proposed, which we evaluate in detail in Chapter 8. We cover top-k strategies proposed by Yang and Stoyanovich [155]: rRD and rND which respectively consider differences and ratios of the proportion of the protected group in top-k compared to the ranking overall, and rKL which uses the KL-divergence to compare distributions of the different groups throughout the ranking. We also include in our study *skew@k* metrics proposed by Geyik *et al.* [70]. This work considers fair ranking in the context of LinkedIn's recruiter platform which helps to identify potential candidates for jobs. The *skew* metrics use the logarithmic ratio of the proportion of the groups in the top-k compared to the overall ranking, as well as a version of the rKL metric. Both works [70, 155] propose post-processing methods to achieve parity in a single ranking according to their respective metrics.

Other methods formulate fair ranking under statistical parity as an optimization problem, without defining an evaluation metric per se. Celis *et al.* [34] formulate fair ranking as a constrained bipartite matching problem, and assume that a user provides fairness rules of the form $L_{kl} \leq G_i \leq U_{kl}$ which set a lower and upper bound on the number of items from each group G_i allowed to appear in the top-k. They solve this using both exact dynamic programming and approximate algorithms, and prove hardness bounds.

In subsequent work [33] the authors consider unfair ranking under a model of implicit bias, and show that upper and lower bound constraints are sufficient to correct for this phenomena. Zehlike *et al.* [161] design a significance test to determine whether representation of items in every prefix of the ranking is fair. The repeated significance tests over dependent subsets of the ranking requires an adjustment to counteract the problem of multiple comparisons. They accomplish this by memoizing a list of probabilities that a fair ranking will be rejected using a significance level α_c. This pre-computed list is used to correct for multiple hypotheses using Šidák's correction. Their method gives a binary fair/not fair assessment, although they suggest this strategy can then be applied as a ranked group fairness measure by using the maximum $\alpha \in [0, 1]$ that a ranking satisfies.

These statistical parity approaches continue to be applied for various settings. For instance, Yang et al. [154] tackle a variation on group fairness in ranking, targeting within-group fairness when there is more than one sensitive data attribute. In [10], Asudeh et al. design pre-processing and database indexing strategies to facilitate fair ranking for real-time applications based on linear scoring.

Pairwise Evaluation Metrics. In [106], we introduce an alternative formulation of statistical parity using pairwise comparisons. We present our *Rpar* metric in Chapter 7 and prove an equivalence between pairwise and top-k statistical parity in Section 9.3.2. In [106] we also use pairwise comparisons to formulate alternative notions of fairness definitions, inspired by the Equalized Odds [80] and predictive parity [39, 98] metrics for classification which compare error-rates across groups to determine fairness. The pairwise formulation has been also been considered by Narasimhan *et al.* [121]. An interesting observation in their work is that pairwise metrics can support continuous group identities. For other proposed formulations, sensitive attributes must be binary (e.g. male / female) or categorical (e.g. different ethnic identities). However since pairwise formulation compare candidates one to one, they can detect unfair scenarios based on a range of values – for instance older candidates favored over younger candidates.

Fairness of Exposure in IR. Other recent work [18, 141, 142, 162] considers fair ranking in Information Retrieval (IR) specifically. Favoring accuracy at the top of a ranking is particularly important in IR where out of possibly thousands of documents returned for a query, only a few top results are likely to be clicked. Rankings are also produced in response to huge numbers of queries, therefore fairness amortized over many results is also a popular formulation. We discuss the exposure based metrics proposed by

Singh and Joachims [141] in detail in Chapter 8 which are based on the expected amount of attention a document will receive in a given rank position. They also propose to use exposure to measure a disparate treatment notion of fairness. Their method targets distributions over rankings using doubly stochastic metrics similar to the assignment matrices we use in our metric comparison framework. They show how rankings can be sampled from such a distribution using the Birkhoff-von Neumann decomposition. Exposure measures are incorporated into a learning-to-rank framework by the authors in later work [142]. Zehlike *et al.* [162] also target a version of the exposure metrics in a fair learning-to-rank method, however they only consider the expected exposure of the top-1 prefix. Concurrent research by Biega *et al.* [18] also considers fair exposure in search rankings, however in this work the authors focus on individual rather than group fairness. This notion of fairness states that similar individuals should be treated similarly [58]. They propose a metric that therefore suggests exposure should be proportional to relevance, when considered in an amortized fashion over many rankings. These ideas are revisited in an analysis of fairness metrics in [54].

Causal Fairness. The variety of ways that fairness in rankings can be measured provides a strong motivation for the in-depth treatment of evaluation metrics in this dissertation. While we are able to demonstrate the efficacy of pairwise approaches for understanding tradeoffs between fairness definitions (Chapter 7), as well as between different ways of evaluating statistical parity (Chapter 8) there are still new approaches being considered as well. Recently much attention has focused on causal fairness, including counterfactual [108] and interventional [96] fairness definitions for classification. As one example, Salimi et al. [135] recently put forth a database repair model in which fairness depends on a causal path from the protected attribute to the outcome through any inadmissible attributes. This analysis aims to understand subtleties inherent in data attributes and causal mechanisms behind unfair bias, and is therefore distinct from our work. For ranking, causal approaches are still in early development, and an exciting direction for future work.

10.2 Comparative Studies of Fairness Metrics

Fairness in rankings has received less attention than other predictive tasks such as classification, despite its critical importance in socio-technical systems. Early work on algorithmic fairness in criminal justice [8] and other aspects of society [12] led to a dearth

of group fairness metrics for classification alongside statistical parity, in particular based on predictive error rates [80]. Theoretical comparative analysis of these fairness metrics yielded important impossibility results showing not all fairness goals can be concurrently satisfied [39, 98]. Subsequent analysis considers tradeoffs between accuracy and fairness [44]. We model our comparative analysis in Chapter 8 after the empirical study of classification fairness metrics by Friedler *et al.* [68]. This work designates three main types of fairness definitions for fair classification and evaluates their interrelationships with a correlation analysis as well as benchmarking on a collection of datasets and different pre-processing techniques.

In contrast to our general comparative analysis of statistical parity metrics, comparative analysis of fair ranking has been focused on IR. Interest in this area is evidenced by the addition in 2019 of a Fair Ranking track at the TREC competition. [1]. Pitoura *et al.* [129] present an overview of types of bias in IR systems categorized as user bias and content bias. User bias occurs when users are shown disparate content - e.g. women are shown lower paid job postings. Content bias is when protected types of content are not given proportional representation. Our comparative study considers the latter type of bias. Combined user-content bias is also suggested to capture echo chamber phenomena where certain groups are only shown certain content. They broadly characterize fairness evaluation in terms of distances between rankings, and point to the need for developing mathematically rigorous fairness metrics as a key challenge.

Other recent work evaluates the applicability different fairness metrics in search and their relationships to evaluation metrics for IR [54, 69]. These works are complementary to ours, demonstrating the relationship of exposure-based strategies for measuring group advantage and other evaluation metrics for ranking using expected fairness over distributions of rankings. In [69] Gao and Shah empirically compare fairness metrics with diversity and novelty metrics. They consider exposure-based statistical parity along with a number of diversity metrics. In [54] Diaz *et al.* propose stochastic distributions of rankings be used as an evaluation framework for evaluating exposure-based fairness metrics. Rather than focus on the statistical parity exposure metrics proposed by Signh and Joachims [141], this work follows an individual parity paradigm similar to that in [18] and considers expected exposure in relation to the relevance of documents and compare these outcomes to traditional IR evaluation metrics.

A major challenge when evaluating fairness metrics is finding real datasets for evaluation, due to the sensitive nature of the data attributes. This often leads to use of

[1] `https://fair-trec.github.io/`

proprietary datasets which cannot be externally validated [70, 136**?**] or use of synthetic or partially synthetic data. In [54] collections of static rankings such as TREC rankings are permuted to produce a distribution. We adopt the bias generation procedure proposed by Yang and Stoyanovich [155] and used in [161] as our standard model group advantage.

10.3 Rank Aggregation

To our knowledge, contemporary fairness criteria had not been applied in the context of rank aggregation prior to our work presented in Chapter 9. In that work we target Kemeny rank aggregation [93] based on the Kendall Tau distance [94], one of the most popular and important aggregation formulations [24]. Kemeny aggregation has enjoyed much interest by the machine learning community. Korba *et al.* [101] develop a statistical learning theory for rank aggregation and provide a recent overview overview of the topic. Applications of Kemeny aggregation range from spam reduction in search results [4, 60], to group recommendation online [11, 14], to biomedical applications [110].

Computing the Kemeny optimal rank aggregation is NP-hard [13, 60]. A full review of methods is beyond the scope of this work. Brancotte *et al.* [23] and Ali and Meila [5] present comparisons of methods for solving Kemeny aggregation along with variations including aggregating partial rankings and allowing ties. However we note the exact integer programming solution of Conitzer et al. [43] and branch-and-bound method by Meila et al. [120] as inspiring our proposed fairness preserving aggregation methods presented in Chapter 9. In their study, Ali and Meila [5] characterize the difficulty of Kemeny aggregation based on agreement among the base rankings in R, which informs our evaluation design in Section 9.5.

Rank aggregation stems from the study of ranked voting in Social Choice Theory [9, 93, 157]. This discipline provides a rich context for asking questions related to contemporary algorithmic fairness. For instance, Chakraborty et al. [35] investigate the application of Social Choice axioms to mitigate the impact of bad actors such as bots on Twitter which may bias recommendations, and to provide fair representation for groups of users with underrepresented preferences. Recent work [25, 32] has explored contemporary fairness for another classic problem in social choice: multi-winner voting. This problem differs from rank aggregation in that only a subset of candidates are selected. Methods proposed are being explored for use in real world voting systems [32]. These examples demonstrate the timely nature of these investigations.

Part III

Conclusion and Future Work

11

Conclusion

In the first part of this dissertation we study the design of automated tools for ranking. Chapter 2 outlines a detailed requirements analysis on key considerations for interactive ranking systems. These considerations provide a roadmap for improving the interactions of users with rankings, so they can better leverage their domain knowledge and intuition to make data driven decisions.

In Chapter 3 guidance is provided on choosing preference collection methods that give the best tradeoff between user effort and training dataset size to allow for useful rankings to be learned from user preferences. Our user study evaluates three preference collection mechanisms and demonstrate the surprising result that user behavior varies greatly across different collection modes. The categorical binning approach is observed to prompt users to organize large amounts of information using broad strokes, providing the most training data to the learning model. The results of these experiments have practical implications for the design of interactive ranking systems, in how best to engage users and derive sufficient information from which to generate meaningful rankings.

In Chapter 4 we present the RanKit system and corresponding "Build, Explore, Explain" paradigm for interactive ranking. Explain features incorporated throughout the system quantify and visualize uncertainty in rank determination to begin addressing requirement 4 and foster trust and understanding in the user, allowing them to effectively collaborate with the system on decision making. Helping users understand rankings is a key goal for future work in this area, which requires meaningful ways of measuring different qualities of rankings.

In the second part of the dissertation we switch gears to this important topic of evaluation of metrics, with a focus on the nascent subject of algorithmic fairness in ranking. In Chapter 7 we present the first methodology for auditing rankings using

pairwise error metrics which capture popular notions of group fairness. We define a set of three new criteria for rankings, which implement fairness definitions previously applied only for fair classification: *Rank Equality, Rank Calibration, and Rank Parity.* Our proposed fairness criteria together with our FARE auditing method comprise a powerful diagnostic tool for nuanced analysis of the treatment of groups being ranked. Then in Chapter 8 we focus on comparing different ways of measuring a single definition of ranking. We present a conceptual framework for comparing statistical parity metrics which measure group advantage in rankings in different ways. We provide guidance on situations where any metric will perform equally well, and certain cases where some metrics may be more appropriate for identifying different types of unfairness.

Finally, Chapter 9 explores fair ranking in a new context of ranking by multiple decision makers. We offer the first formulation of the fair rank aggregation problem as an extension of Kemeny aggregation integrated with contemporary group fairness criteria. We show how our pairwise metrics can be applied to facilitate integrated solutions to the fair rank aggregation problem. A rich family of exact and approximate algorithms are presented which solve this new optimization problem by enforcing statistical parity fairness semantics. Examples of real-world scenarios we consider are hiring by committee and aggregating rankings created by sports analysts. For strict fairness requirements, our exact fairness-aware ILP methods are robust to different amounts of bias agreement among the base rankings for rankings with $n < 100$ candidates. However this approach suffers from a high time complexity. When fairness requirements are lenient, our fairness-aware B+B solution speeds runtime considerably while achieving optimal aggregation results. We find that enforcing statistical parity fairness semantics using this approach is very sensitive to disagreement in the base rankings and strict fairness requirements. Lastly, our approximate fair post-processing algorithm guarantees fairness while introducing the minimal amount of approximation error.

Together the work presented in these two parts expands the ways in which ranking can support meaningful decision making. Automating such processes is becoming evermore common, therefore tools and techniques to help people understand the basis for and implications of these outcomes are essential. We next discuss many interesting avenues for further discovery in this area.

12

Discussion and Future Directions

Visual explanations can play a key role in facilitating fair applications of machine learning in our society, however they must be integrated into usable tools that people can deploy using their own data. Fairness assessment algorithms can do little to mitigate the risks of automated decision making if they are not understandable to the decision makers or those impacted by the decisions. Interactive data visualization provides a way for experts and end users alike to see and understand the influence of sensitive data attributes on decision outcomes.

Much recent work investigates this exciting new area. Questions include the effectiveness of providing explanations for automated decisions [19, 56], transparency of decision making to users [163], and whether collaboration with automated predictions helps analysts make more objective decisions [73]. Early research is beginning to evaluate the relationship between automated fairness criteria and human perceptions of fairness [128, 137, 143]. Crowdsourced studies by Srivastava *et al.* [143] and Saxena *et al.* [137] evaluate which fairness metrics match people's intuitions of fairness for classification tasks. Such direct evaluation has not yet been performed for fairness in ranking problems, although Peng *et al.* [128] have evaluated how fairly human analysts rank members of different groups when given different proportions of candidates. Other research investigates the fairness of ranked search results for hiring websites [37, 70, 78].

Building on the ideas developed in this dissertation for interactive ranking, evaluating fairness in rankings, and fair consensus ranking, we now outline three interesting topics for further study:

12.1 Visualizing Fairness in Rankings

In the study of algorithmic fairness, we are interested in identifying and mitigating any unfair bias inherent in the data and algorithmic systems that are used for decision making in people's lives [12]. Fairness can take on many different meanings in this context as we have discussed throughout part two of this dissertation. Many distinctions are considered such as between individual fairness and group fairness, ways of measuring group advantage, fairness in a single instance or amortized over a distribution of outcomes, and fair decisions made by a single model or by consensus among a group. Other extremely important questions around fairness among multiple overlapping group identities unfortunately are outside the scope of this work like much scholarship on fairness.

Designing visual interactions to communicate these ideas is a challenging direction for future work. Communicating these complicated ideas is critical for the adoption of fairness-preserving technologies, driven by diverse coalitions of technical and non-technical stakeholders. A compelling approach has been to use combinations of text and different visual designs to communicate ideas using visual storytelling. For instance, Google's PAIR lab published an interactive visualization [1] illustrating the issue of group fairness for lending, inspired by the notion of error-based fairness called Equalized Odds given by Hardt *et al.* [80]. Following from this example of visual storytelling, similar approaches to visualizing unfairness in rankings. However, visually representing rankings still poses a number of challenges. As we know from studying IR systems, only so many search results can be represented on a page, and users typically only interact with a small set of items at the top on the list, rarely exploring past the first set of results. This may be fine if searching for a single item, however this presentation does not lend itself to understanding an entire ranking. Another consideration is that for meaningful decision making, each item typically has associated information or data attributes, and organizing this information into a federated view can be difficult. We discuss visualization of rankings in interactive systems in Section 5. Further, many key ideas must be communicated regarding fairness in rankings including:

- Ranked order of items.
- Data attributes associated with each item being ranked.
- Holistic *and* fine-grained representations of the ranking.
- Group membership of each item.
- Balance of groups in total population.

[1] `https://research.google.com/bigpicture/attacking-discrimination-in-ml`

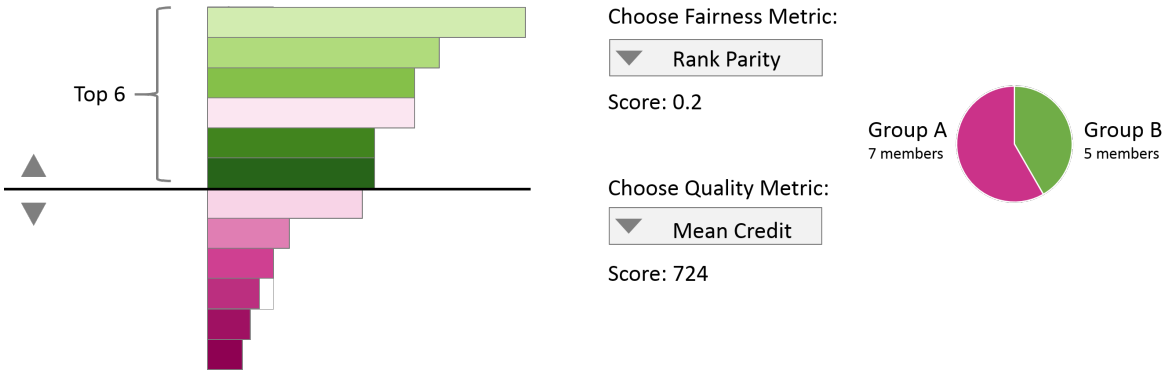


Figure 12.1: A mockup of a fair ranking dashboard.

- Preferred outcome for items (e.g. membership in the top-k prefix of the ranking).
- Alternative scenarios (e.g. "what-if" analysis that shows impact of adjustments to the ranking).
- Definitions of fairness.
- Evaluation metrics for rank quality and fairness.

Figure 12.1 mocks up a version of this kind of visual storytelling for a simple fair ranking scenario for a ranking of 12 items over two groups. Group status is indicated with color, and intensity of the color indicates rank position. The width of the stacked bars in the histogram represent an underlying scoring function. Scores are not required for all rankings, however many fairness evaluation approaches consider trade offs between utility and fairness. Therefore visualizing scores can allow the user to see when rank accuracy is violated. A pie chart gives the proportion of items in each group, and evaluation metrics are accessed using drop-down lists.

This type of display could be integrated with a number of interactions to facilitate understanding. For instance, users could adjust the slider to consider only a top prefix of the ranking. The histogram provides something of a "snapshot" view, which could provide additional information about the items on hover, or users could click through to access a full table view. Finally, bias mitigation methods could be incorporated to allow the user to see how the ranking would change if it were adjusted to be more fair, perhaps using animation to rearrange the bars in the histogram.

In this mockup the actual unfairness in the ranking is only communicated by the fairness score (and the impression given by the distribution of the colors throughout the list). An alternative approach would be to *visualize the bias itself directly.* In the FARE framework (Section 7.2) we explored this idea by creating plots of the error for each

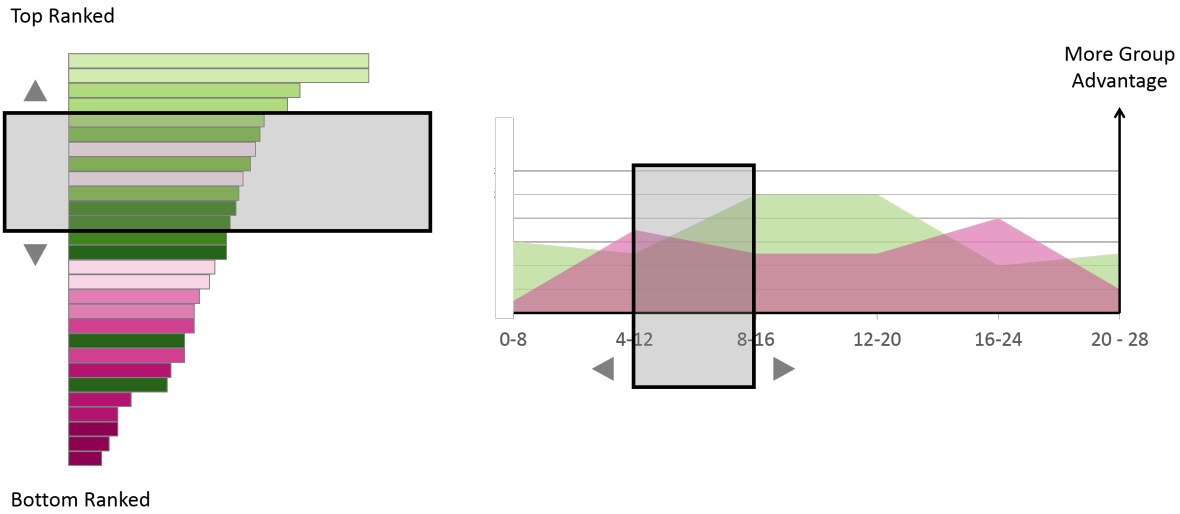


Figure 12.2: Mockup of a sliding window interaction visualizing group fairness in a single ranking.

group. We applied our fair rank measurements using a sliding window approach to show the degree of advantage for each group at different rank positions. Figure 12.2 shows a version of this sliding window strategy incorporated into a rank snapshot view. On the left is a vertical representation of a ranking, and on the right the fairness score for each group is plotted. The x-axis of the fairness chart corresponds to the window position, and the y-axis corresponds to a measure of group advantage. Users can interact by sliding the grey box on either element to view the corresponding rank range on the other element.

12.2 Visual Interactive Support for Fair Consensus Ranking

In Chapter 9 we present the fair rank aggregation problem along with algorithms for aggregating multiple rankings into a single consensus ranking which preserves or ensures group-fair outcomes. This technology could assist decision makers for tasks such as the hiring scenario above, by automatically combining the rankings of a hiring committee into a single ranking which balances the preferences of each decision maker with group fairness considerations for the candidates being ranked. Interactive interfaces incorporated into mixed initiative systems for consensus could play a crucial role in allowing people to use this technology for real world decision making. The careful design of interactions should empower decision makers to understand and trust the algorithmic process, answering

questions throughout the consensus building process such as:

- Are my preferences being represented well in the consensus?
- How fair is the set of base rankings?
- To what extent do the decision makers agree on the consensus ranking?
- Where in the rankings are trade-offs for fairness being made?

Visualizing the Committee Rankings. To support a consensus-building process a key consideration is visualising the set of proposed rankings to be aggregated, as well as the set of possible consensus outcomes. In contrast to the strategies I discussed above this involves capturing the interrelationships among possibly a large number of rankings to be considered.

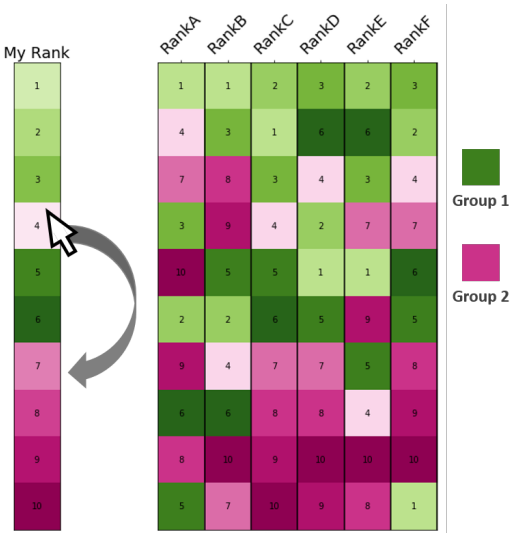


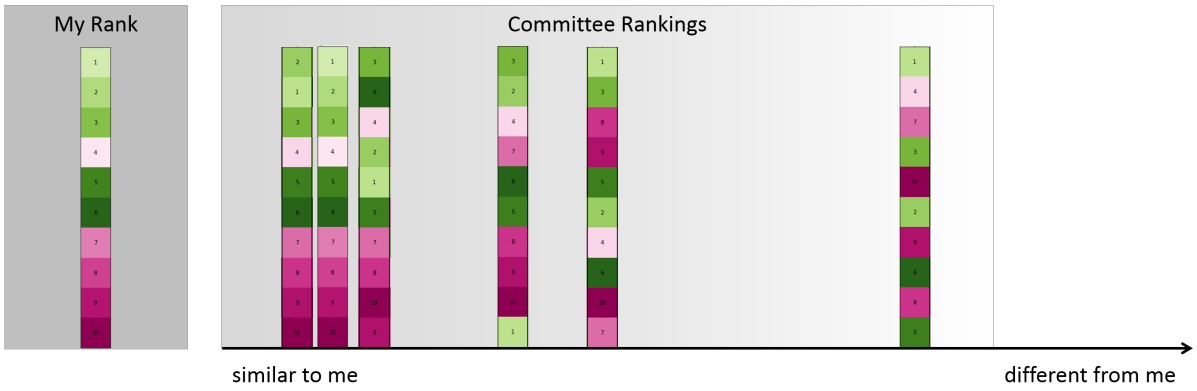
Figure 12.3: Mock up of an interactive visualization for comparing rankings.

Figure 12.3 shows a simple representation of a set of rankings. Such a design could incorporate drag and drop interactions which allow users to swap items in their own ranking and see how the consensus ranking changes, understanding the impact of their personal preferences on the outcome. Multiple metrics could be supported as was shown in the dashboard in Figure 12.1 to give information to the user about fairness, accuracy of the aggregation, and to otherwise characterize the set of input rankings. Hopefully such feedback can help users understand the fairness of their decisions, encourage self-reflection and discussion among users, and assist the decision makers in iteratively adjusting their base rankings and the imposed fairness constraints, until they find a satisfactory outcome for their decision making task.

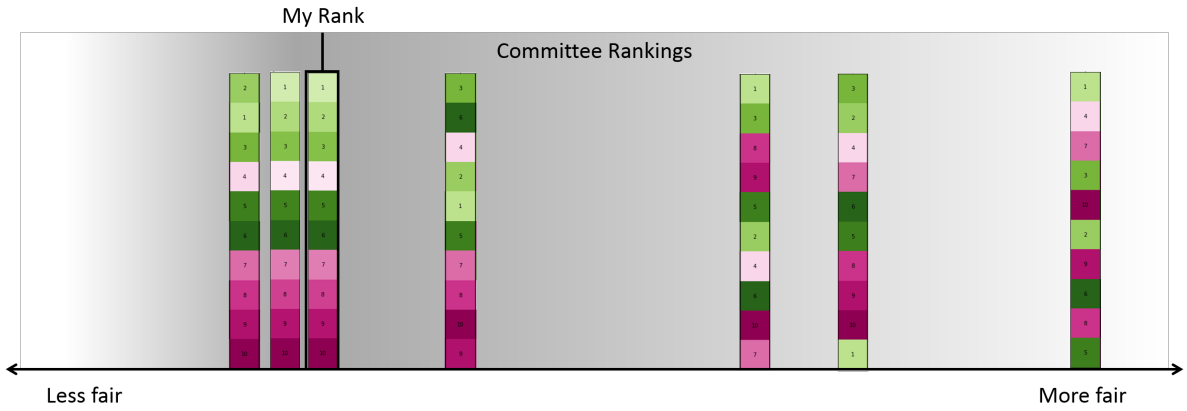
Visualizing Relationships between Many Rankings. A drawback to the visualization in Figure 12.3 is that it doesn't express any relationships among the rankings in the set, aside from color patterns that may catch a user's eye. Rankings are combinatorial objects and the scale of the space of possible rankings over a set of items quickly explodes as the number of candidate items being ranked increases. Visually representing such a space is a challenging task that long captured many imaginations. For instance, consider the use of Cayley diagrams to represent group structure in set theory, or the study of the permutation polytope [156]. These strategies attempt conceptualize groups of permutations as geometric objects, but they do not lend themselves to visualizing rankings over more than four items. Clearly, for today's information needs, new approaches must be developed.

One approach is to treat rankings a high dimensional vectors and perform dimensionality reduction to represent the rankings in a 2D space. Kidwell *et al.* [95] for instance use the Kendall Tau distance as basis for multidimensional scaling to embed rankings in a 2d space. Clustering is then performed over the rankings rankings and visualized using heatmaps. Other dimensionality reduction strategies popular for visualization such as t-SNE [115] might also be adapted for this purpose. However, a drawback could be that the resulting visualizations may not align with usual notions of what a ranking is, or typical tools (e.g. spreadsheets) used to analyse them. Familiarity has been shown to promote trust in users [48], and therefore this approach may not be ideal for understanding relationships between rankings.

As an alternative approach could leverage the fact that for a single user of a consensus building tool, they are not necessarily considering the entire space of possible rank outcomes. Rather, their analysis is anchored in their own set of preferences used to create their ranking. They are most likely trying to understand how the other rankings in the system are related *to their own ranking*, and how any changes might impact those relationships. Therefore we can simply visualize the distances to the other rankings, instead of having to consider every relationship between every ranking. As shown in figure 12.4a this can be done in a straightforward horizontal layout with the user's ranking on the left and the other rankings arranged by increasing dissimilarity to the right. The distances between rankings can be determined using any number of rank comparison metrics including the Kendall Tau distance [94]. This can be incorporated easily with other dashboard elements and interactions. Different versions switch the anchor ranking that the visualization is centered on, and evaluate other metrics in addition to distance between rankings. Figure 12.4b uses the same strategy to compare the fairness of the



(a) The user's ranking is shown on the left, with the other committee rankings organized by increasing dissimilarity to the right.



(b) Committee rankings are arranged from least fair to most fair.

Figure 12.4: Mockup of interactive visual comparisons for a rank consensus ranking task.

committee rankings.

12.3 Understanding Human Perceptions of Fairness

Increasingly, the judgments of human analysts are augmented by decision support tools or even fully automated screening procedures which automatically rank candidates. A number of recent studies have focused on auditing the fairness of real-world ranked search results for tasks such as hiring [37, 70, 78]. When we design new tools along this vein, such as those I have brainstormed here, principled inquiry must be employed in the development and design process. In [85] Holstein *et al.* survey machine learning practitioners to investigate challenges and roadblocks in designing fair technologies and identifying practical approaches to building fair technologies. Binns et al. study the effectiveness of

providing explanations for automated decisions in the context of compliance with new regulations in the GDPR [19]. The need for transparency of the decision making procedures and algorithms is commonly touted, but the required degree to which automated systems must be transparent is still being debated, as considered by Zerilli *et al.* [163].

Most pertinent to the designs presented here is whether automated systems can effectively help people make more fair decisions. Recent studies have tried to understand this from different angles. For instance, Green *et al.* studied automated risk assessments for recidivism prediction and attempted to understand whether it makes human decisions more objective [73]. Their work suggests that there is much more study that must be done to ensure this. Other research is just beginning to evaluate the relationship between proposed measures of algorithmic fairness and human perceptions of fairness [128, 137, 143]. Crowdsourced studies by Srivastava *et al.* [143] and Saxena *et al.* [137] evaluate which fairness metrics match people's intuitions of fairness for classification tasks. Such direct evaluation has not yet been performed for fairness in ranking problems. One step is this direction is a study by Peng *et al.* [128] which evaluated how fairly human analysts rank members of different groups when presented with different proportions of candidates.

Basic building blocks are required to initiate rigorous a visualization design and evaluation loop to evaluate people's understanding of fairness in rankings. Approaches should facilitate understanding of features that align with decision-makers' personal values and understanding of fairness, as well as fixed legal and ethical constraints. For example, a survey asking people to compare two rankings and choose the one that is more fair is a simple approach to beginning investigation, which could reveal complex insights into people's perceptions of fairness. These impressions could be compared against fairness metrics considered in this dissertation to evaluate whether they align with people's impressions. Another study could be modeled after [73] by asking people to choose fair rankings assisted by automatically generated evaluation metrics. Study details should be developed through an iterative process using prototype systems and small-scale pilot experiments. These questions are as of now not well understood, and such preliminary investigation is needed to further elucidate them.

References

[1] E. Agapie, G. Golovchinsky, and P. Qvarfordt. Leading people to longer queries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3019–3022. ACM, 2013. 23

[2] C. C. Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016. 6, 33, 34

[3] I. Ajunwa, S. Friedler, C. E. Scheidegger, and S. Venkatasubramanian. Hiring by algorithm: predicting and preventing disparate impact. 2016. 1

[4] L. Akritidis, D. Katsaros, and P. Bozanis. Effective rank aggregation for metasearching. *Journal of Systems and Software*, 84(1):130–143, 2011. 112

[5] A. Ali and M. Meilă. Experiments with kemeny ranking: What works when? *Mathematical Social Sciences*, 64(1):28–40, 2012. 75, 77, 82, 99, 100, 112

[6] K. M. Altenburger, R. De, K. Frazier, N. Avteniev, and J. Hamilton. Are there gender differences in professional self-promotion? an empirical case study of linkedin profiles among recent mba graduates. In *Eleventh International AAAI Conference on Web and Social Media*, 2017. 1

[7] X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver. Rate it again: increasing recommendation accuracy by user re-rating. In *Proceedings of the third ACM conference on Recommender systems*, pages 173–180. ACM, 2009. 9, 35

[8] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. *Pro Publica*, 2016. 53, 110

[9] K. J. Arrow. *Social choice and individual values*. Yale University Press, 1963. 76, 112

[10] A. Asudeh, H. Jagadish, J. Stoyanovich, and G. Das. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1259–1276. ACM, 2019. 1, 2, 57, 76, 77, 109

[11] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126. ACM, 2010. 112

[12] S. Barocas and A. D. Selbst. Big data's disparate impact. *Cal. L. Rev.*, 104:671, 2016. 1, 38, 110, 117

[13] J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare*, 6(2):157–165, 1989. 76, 77, 86, 112

[14] J. P. Baskin and S. Krishnamurthi. Preference aggregation in group recommender systems for committee decision-making. In *Proceedings of the third ACM conference on Recommender systems*, pages 337–340. ACM, 2009. 112

[15] T. Berg, V. Burg, A. Gombović, and M. Puri. On the rise of fintechs–credit scoring using digital footprints. Technical report, National Bureau of Economic Research, 2018. 1

[16] M. Bertrand and S. Mullainathan. Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination. *American economic review*, 94(4):991–1013, 2004. 1

[17] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi, and C. Goodrow. Fairness in recommendation ranking through pairwise comparisons. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2212–2220, New York, NY, USA, 2019. ACM. 59

[18] A. J. Biega, K. P. Gummadi, and G. Weikum. Equity of attention: Amortizing individual fairness in rankings. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 405–414. ACM, 2018. 2, 39, 40, 109, 110, 111

[19] R. Binns, M. Van Kleek, M. Veale, U. Lyngs, J. Zhao, and N. Shadbolt. 'it's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions.

In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 377. ACM, 2018. 116, 123

[20] I. Bohnet, A. Van Geen, and M. Bazerman. When performance trumps gender bias: Joint vs. separate evaluation. *Management Science*, 62(5):1225–1234, 2015. 1

[21] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016. 38

[22] J. Boy, F. Detienne, and J.-D. Fekete. Storytelling in information visualizations: Does it engage users to explore data? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1449–1458. ACM, 2015. 11

[23] B. Brancotte, B. Yang, G. Blin, S. Cohen-Boulakia, A. Denise, and S. Hamel. Rank aggregation with ties: Experiments and analysis. *Proceedings of the VLDB Endowment*, 8(11):1202–1213, 2015. 75, 82, 88, 100, 104, 112

[24] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016. 75, 76, 82, 112

[25] R. Bredereck, P. Faliszewski, A. Igarashi, M. Lackner, and P. Skowron. Multiwinner elections with diversity constraints. In *AAAI Conference on Artificial Intelligence*, 2018. 112

[26] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018. 38, 41

[27] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96. ACM, 2005. 33

[28] A. Calero Valdez, M. Ziefle, and K. Verbert. Hci for recommender systems: the past, the present and the future. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 123–126. ACM, 2016. 34

[29] G. Carenini and J. Loyd. Valuecharts: analyzing linear models expressing preferences and evaluations. In *Proceedings of the working conference on Advanced visual interfaces*, pages 150–157. ACM, 2004. 33, 35

[30] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there. In *European Conference on Information Retrieval*, pages 16–27. Springer, 2008. 8, 12, 35

[31] R. Caruana and A. Niculescu-Mizil. Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–78. ACM, 2004. 46, 49

[32] L. E. Celis, L. Huang, and N. K. Vishnoi. Multiwinner voting with fairness constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 144–151. AAAI Press, 2018. 112

[33] L. E. Celis, A. Mehrotra, and N. K. Vishnoi. Interventions for ranking in the presence of implicit bias. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 369–380, 2020. 38, 70, 109

[34] L. E. Celis, D. Straszak, and N. K. Vishnoi. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017. 2, 39, 40, 46, 57, 76, 77, 108

[35] A. Chakraborty, G. K. Patro, N. Ganguly, K. P. Gummadi, and P. Loiseau. Equality of voice: Towards fair representation in crowdsourced top-k recommendations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 129–138. ACM, 2019. 112

[36] T. M. Chan and M. Pătraşcu. Counting inversions, offline orthogonal range counting, and related problems. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 161–173. Society for Industrial and Applied Mathematics, 2010. 50

[37] L. Chen, R. Ma, A. Hannák, and C. Wilson. Investigating the impact of gender on rank in resume search engines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 651. ACM, 2018. 1, 38, 116, 122

[38] L. Chen and P. Pu. Survey of preference elicitation methods. Technical report, 2004. 8, 12, 35

[39] A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017. 38, 56, 109, 111

[40] J. Clausen. Branch and bound algorithms-principles and examples. 1999. 92

[41] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. In *Advances in Neural Information Processing Systems*, pages 451–457, 1998. 76

[42] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994. 34

[43] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing kemeny rankings. In *AAAI Conference on Artificial Intelligence*, volume 6, pages 620–626, 2006. 76, 86, 87, 88, 98, 99, 105, 112

[44] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806. ACM, 2017. 2, 38, 111

[45] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *Advances in neural information processing systems*, pages 313–320, 2004. 40

[46] R. J. Crouser, L. Franklin, A. Endert, and K. Cook. Toward theoretical techniques for measuring the use of human effort in visual analytic systems. *IEEE transactions on visualization and computer graphics*, 23(1):121–130, 2017. 6, 8, 11

[47] G. Cumming. *Understanding the new statistics: Effect sizes, confidence intervals, and meta-analysis*. Routledge Publishing, 2013. 17, 18

[48] A. Dasgupta, J.-Y. Lee, R. Wilson, R. A. Lafrance, N. Cramer, K. Cook, and S. Payne. Familiarity vs trust: A comparative study of domain scientists' trust in visual analytics and conventional analysis methods. *IEEE transactions on visualization and computer graphics*, 23(1):271–280, 2016. 121

[49] J. Dastin. Amazon scraps secret ai recruiting tool that showed bias against women. *Reuters*, 2018. 1

[50] J. C. de Borda. Memoire sur les elections au scrutin, 1781. *Histoire de l'Academie Royale des Sciences, Paris*, 1953. 99

[51] R. DeVol, J. Lee, and M. Ratnatunga. 2016 State Technology and Science Index: Sustaining america's innovation economy. 2016. 5

[52] P. Diaconis. Group representations in probability and statistics. *Lecture Notes-Monograph Series*, 11:i–192, 1988. 43

[53] P. Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 262–268, 1977. 43

[54] F. Diaz, B. Mitra, M. D. Ekstrand, A. J. Biega, and B. Carterette. Evaluating stochastic rankings with expected exposure, 2020. 41, 110, 111, 112

[55] E. Dimara, A. Bezerianos, and P. Dragicevic. Narratives in crowdsourced evaluation of visualizations: A double-edged sword? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2017. 11

[56] J. Dodge, Q. V. Liao, Y. Zhang, R. K. Bellamy, and C. Dugan. Explaining models: an empirical study of how explanations impact fairness judgment. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 275–285. ACM, 2019. 116

[57] P. Dragicevic. Fair statistical communication in hci. In *Modern Statistical Methods for HCI*, pages 291–330. Springer, 2016. 18

[58] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226. ACM, 2012. 39, 56, 108, 110

[59] C. Dwork, N. Immorlica, A. T. Kalai, and M. D. Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133, 2018. 38

[60] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001. 43, 76, 77, 86, 112

[61] T. U. EEOC. Uniform guidelines on employee selection procedures. 1979. 1

[62] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 473–482. ACM, 2012. 34

[63] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 47–58. ACM, 2004. 75, 82

[64] M. Falahatgar, Y. Hao, A. Orlitsky, V. Pichapati, and V. Ravindrakumar. Maxing and ranking with few assumptions. In *Advances in Neural Information Processing Systems*, pages 7063–7073, 2017. 7, 34

[65] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015. 38, 39, 53, 76

[66] M. Feng, C. Deng, E. M. Peck, and L. Harrison. Hindsight: Encouraging exploration through direct encoding of personal interaction history. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):351–360, 2017. 11

[67] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003. 33

[68] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 329–338. ACM, 2019. 111

[69] R. Gao and C. Shah. Toward creating a fairer ranking in search engine results. *Information Processing & Management*, 57(1):102138, 2020. 111

[70] S. C. Geyik, S. Ambler, and K. Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2221–2231. ACM, 2019. 1, 2, 38, 39, 40, 57, 58, 59, 76, 77, 106, 108, 112, 116, 122

[71] M. Gladwell. The order of things. *The New Yorker*, 87(1):68–75, 2011. 2, 5

[72] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit. Lineup: Visual analysis of multi-attribute rankings. *IEEE transactions on visualization and computer graphics*, 19(12):2277–2286, 2013. 5, 7, 23, 33, 35

[73] B. Green and Y. Chen. Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 90–99. ACM, 2019. 116, 123

[74] A. G. Greenwald and M. R. Banaji. Implicit social cognition: attitudes, self-esteem, and stereotypes. *Psychological review*, 102(1):4, 1995. 1

[75] Z. Guan and E. Cutrell. An eye tracking study of the effect of target rank on web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 417–420. ACM, 2007. 8

[76] L. Gurobi Optimization. Gurobi optimizer reference manual, 2019. 88, 99

[77] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982. 84, 85

[78] A. Hannák, C. Wagner, D. Garcia, A. Mislove, M. Strohmaier, and C. Wilson. Bias in online freelance marketplaces: Evidence from taskrabbit and fiverr. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 1914–1933. ACM, 2017. 38, 116, 122

[79] K. Hara, A. Adams, K. Milland, S. Savage, C. Callison-Burch, and J. P. Bigham. A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 449. ACM, 2018. 15

[80] M. Hardt, E. Price, N. Srebro, et al. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016. 38, 39, 108, 109, 111, 117

[81] S. Haroz, R. Kosara, and S. L. Franconeri. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1191–1200. ACM, 2015. 11

[82] C. He, D. Parra, and K. Verbert. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56:9–27, 2016. 6, 12, 34, 35

[83] R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. 1999. 33, 40

[84] P. Hofmann. Statlog (german credit data) data set. *UCI Repository of Machine Learning Databases*, 1994. 53

[85] K. Holstein, J. Wortman Vaughan, H. Daumé III, M. Dudik, and H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 600. ACM, 2019. 122

[86] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM, 1999. 6, 7, 33

[87] W. House. College scorecard. 2013. 5

[88] R. Hu and P. Pu. Helping users perceive recommendation diversity. In *DiveRS@ RecSys*, pages 43–50, 2011. 35

[89] M. Jacob, B. Kimelfeld, and J. Stoyanovich. A system for management and analysis of preference data. *Proceedings of the VLDB Endowment*, 7(12):1255–1258, 2014. 75, 82

[90] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. 53, 58

[91] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002. 8, 12, 14, 33

[92] T. Joachims, L. A. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Sigir*, volume 5, pages 154–161, 2005. 40

[93] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959. 75, 76, 78, 79, 82, 112

[94] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. 43, 59, 82, 112, 121

[95] P. Kidwell, G. Lebanon, and W. Cleveland. Visualizing incomplete and partially ranked data. *IEEE Transactions on visualization and computer graphics*, 14(6), 2008. 36, 121

[96] N. Kilbertus, M. R. Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf. Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems*, pages 656–666, 2017. 110

[97] J. Kleinberg and M. Raghavan. Selection problems in the presence of implicit bias. *Proceedings of the 9th Conference on Innovations in Theoretical Computer Science*, 2018. 70

[98] J. M. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Proceedings of the 8th Conference on Innovation in Theoretical Computer Science*, page 43:1–43:23, 2017. 38, 109, 111

[99] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):441–504, 2012. 6, 8, 34

[100] J. A. Konstan and J. Riedl. Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction*, 22(1-2):101–123, 2012. 6, 34

[101] A. Korba, S. Clémençon, and E. Sibony. A learning theory of ranking aggregation. In *Artificial Intelligence and Statistics*, pages 1001–1010, 2017. 112

[102] C. Kuhlman, D. Doherty, M. Nurbekova, G. Deva, Z. Phyo, P.-H. Schoenhagen, M. VanValkenburg, E. Rundensteiner, and L. Harrison. Evaluating preference collection methods for interactive ranking analytics. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 512. ACM, 2019. 10

[103] C. Kuhlman, D. Doherty, M. Nurbekova, G. Deva, Z. Phyo, P.-H. Schoenhagen, M. VanValkenburg, E. Rundensteiner, and L. Harrison. Evaluating preference collection methods for interactive ranking analytics. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 203–212. ACM, 2019. 26

[104] C. Kuhlman, E. Rundensteiner, R. Neamtu, R. Ahsan, J. Stokes, A. Hoxha, J. Bao, S. Gvozdenovic, T. Meyer, N. Patel, et al. Towards an interactive learn-to-rank system for economic competitiveness understanding. In *KDD 2017 Interactive Data Exploration and Analytics Workshop*, 2017. 8, 23, 25, 33, 34

[105] C. Kuhlman, M. VanValkenburg, D. Doherty, M. Nurbekova, G. Deva, Z. Phyo, E. Rundensteiner, and L. Harrison. Preference-driven interactive ranking system for personalized decision support. In *Proceedings of the International Conference on Information and Knowledge Management*. ACM, 2018. 5, 8, 12, 23, 33

[106] C. Kuhlman, M. VanValkenburg, and E. Rundensteiner. Fare: Diagnostics for fair ranking using pairwise error metrics. In *The World Wide Web Conference*, pages 2936–2942. ACM, 2019. 2, 39, 40, 43, 59, 76, 78, 82, 109

[107] R. Kumar and S. Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th International Conference on World Wide Web*, pages 571–580. ACM, 2010. 43

[108] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4066–4076, 2017. 110

[109] I. Li, A. Dey, and J. Forlizzi. A stage-based model of personal informatics systems. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 557–566. ACM, 2010. 7

[110] S. Lin. Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570, 2010. 76, 112

[111] T.-Y. Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009. 5, 8, 33, 40

[112] B. Loepp, K. Herrmanny, and J. Ziegler. Blended recommending: Integrating interactive information filtering and algorithmic recommender techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 975–984. ACM, 2015. 24, 35

[113] B. Loepp, T. Hussein, and J. Ziegler. Choice-based preference elicitation for collaborative filtering recommender systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3085–3094. ACM, 2014. 12, 35

[114] B. Long, O. Chapelle, Y. Zhang, Y. Chang, Z. Zheng, and B. Tseng. Active learning for ranking through expected loss optimization. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274. ACM, 2010. 34

[115] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 121

[116] C. L. Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957. 78, 100

[117] B. Mandhani and M. Meila. Tractable search for learning exponential models of rankings. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 392–399. PMLR, 2009. 99, 100

[118] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947. 84

[119] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, and F. Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 2243–2251. ACM, 2018. 39

[120] M. Meila, K. Phadnis, A. Patterson, and J. A. Bilmes. Consensus ranking under the exponential model. In *Proceedings of the 23rd Annual Conference on Uncertainty in Artificial Intelligence*, pages 285–294, 2007. 76, 89, 92, 98, 104, 112

[121] H. Narasimhan, A. Cotter, M. Gupta, and S. Wang. Pairwise fairness for ranking and regression. *arXiv preprint arXiv:1906.05330*, 2019. 40, 59, 109

[122] Z. Obermeyer and S. Mullainathan. Dissecting racial bias in an algorithm that guides health decisions for 70 million people. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 89–89. ACM, 2019. 2

[123] A. Olteanu, J. Garcia-Gathright, M. de Rijke, and M. D. Ekstrand. Facts-ir: Fairness, accountability, confidentiality, transparency, and safety in information retrieval. 38

[124] C. O'Neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books, 2017. 2, 38

[125] T. Pahikkala, E. Tsivtsivadze, A. Airola, J. Boberg, and T. Salakoski. Learning to rank with pairwise regularized least-squares. In *SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, volume 80, pages 27–33. Citeseer, 2007. 33

[126] S. Pajer, M. Streit, T. Torsney-Weir, F. Spechtenhauser, T. Möller, and H. Piringer. Weightlifter: Visual weight space exploration for multi-criteria decision making. *IEEE transactions on visualization and computer graphics*, 23(1):611–620, 2017. 33, 35, 36

[127] J. Pearl. Heuristics: intelligent search strategies for computer problem solving. 1984. 88

[128] A. Peng, B. Nushi, E. Kıcıman, K. Inkpen, S. Suri, and E. Kamar. What you see is what you get? the impact of representation criteria on human bias in hiring. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 125–134, 2019. 116, 123

[129] E. Pitoura, P. Tsaparas, G. Flouris, I. Fundulaki, P. Papadakos, S. Abiteboul, and G. Weikum. On measuring bias in online information. *ACM SIGMOD Record*, 46(4):16–21, 2018. 111

[130] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5680–5689, 2017. 38

[131] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 570–579. ACM, 2007. 27, 29, 34

[132] A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning*, pages 118–126, 2014. 7, 34

[133] R. E. Robertson, D. Lazer, and C. Wilson. Auditing the personalization and composition of politically-related search engine results pages. In *Proceedings of the 2018 World Wide Web Conference*, pages 955–965, 2018. 41

[134] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim. The role of uncertainty, awareness, and trust in visual analytics. *IEEE transactions on visualization and computer graphics*, 22(1):240–249, 2016. 6, 9

[135] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*, pages 793–810, 2019. 1, 77, 110

[136] P. Sapiezynski, W. Zeng, R. E. Robertson, A. Mislove, and C. Wilson. Quantifying the impact of user attention on fair group representation in ranked lists. *arXiv preprint arXiv:1901.10437*, 2019. 41, 112

[137] N. A. Saxena, K. Huang, E. DeFilippis, G. Radanovic, D. C. Parkes, and Y. Liu. How do fairness definitions fare?: Examining public attitudes towards algorithmic definitions of fairness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 99–106. ACM, 2019. 116, 123

[138] F. Schalekamp and A. v. Zuylen. Rank aggregation: Together we're strong. In *2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments*, pages 38–51. SIAM, 2009. 76, 86

[139] K. Schwab, X. Sala-i Martin, et al. The global competitiveness report 2016-2017. World Economic Forum, 2017. 5

[140] C. Shi, W. Cui, S. Liu, P. Xu, W. Chen, and H. Qu. Rankexplorer: Visualization of ranking changes in large time series data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2669–2678, 2012. 33

[141] A. Singh and T. Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2219–2228. ACM, 2018. 2, 38, 39, 40, 41, 59, 109, 110, 111

[142] A. Singh and T. Joachims. Policy learning for fairness in ranking. *arXiv preprint arXiv:1902.04056*, 2019. 109, 110

[143] M. Srivastava, H. Heidari, and A. Krause. Mathematical notions vs. human perception of fairness: A descriptive approach to fairness for machine learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2459–2468, New York, NY, USA, 2019. ACM. 116, 123

[144] M. Szummer and E. Yilmaz. Semi-supervised learning to rank with preference regularization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 269–278. ACM, 2011. 7, 14, 34

[145] E. L. Uhlmann and G. L. Cohen. Constructed criteria: Redefining merit to justify discrimination. *Psychological Science*, 16(6):474–480, 2005. 1

[146] A. Waldman and S. Wei. Colleges flush with cash saddle poorest students with debt. *Pro Publica*, 2015. 2

[147] E. Wall, S. Das, R. Chawla, B. Kalidindi, E. T. Brown, and A. Endert. Podium: Ranking data using mixed-initiative visual analytics. *IEEE transactions on visualization and computer graphics*, 24(1):288–297, 2018. 5, 8, 9, 12, 23, 24, 33, 34, 35

[148] J. Walny, S. Huron, C. Perin, T. Wun, R. Pusch, and S. Carpendale. Active reading of visualizations. *IEEE transactions on visualization and computer graphics*, 24(1):770–780, 2018. 11

[149] R. L. Wasserstein and N. A. Lazar. The asa's statement on p-values: Context, process, and purpose. *The American Statistician*, 2016. 17

[150] A. Waters and R. Miikkulainen. Grade: Machine learning support for graduate admissions. *AI Magazine*, 35(1):64–64, 2014. 1, 38, 75

[151] F. Wauthier, M. Jordan, and N. Jojic. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning*, pages 109–117, 2013. 6, 7, 8, 14, 34

[152] Y. Wu, L. Zhang, and X. Wu. On discrimination discovery and removal in ranked data using causal graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2536–2544, 2018. 39

[153] Y. Wu, L. Zhang, and X. Wu. On discrimination discovery and removal in ranked data using causal graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '18, page 2536–2544, New York, NY, USA, 2018. Association for Computing Machinery. 77

[154] K. Yang, V. Gkatzelis, and J. Stoyanovich. Balanced ranking with diversity constraints. *arXiv preprint arXiv:1906.01747*, 2019. 109

[155] K. Yang and J. Stoyanovich. Measuring fairness in ranked outputs. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 22:1–22:6. ACM, 2017. 2, 39, 40, 46, 57, 58, 72, 76, 77, 78, 106, 108, 112

[156] V. Yemelicher, M. M. Kovalev, M. Dravtsov, and G. Lawden. *Polytopes, graphs and optimisation*. Cambridge University Press, 1984. 36, 121

[157] P. Young. Optimal voting rules. *Journal of Economic Perspectives*, 9(1):51–64, 1995. 76, 100, 112

[158] H. Yu. Svm selective sampling for ranking with application to data retrieval. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 354–363. ACM, 2005. 34

[159] Y. Yue, R. Patel, and H. Roehrig. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th international conference on World wide web*, pages 1011–1018. ACM, 2010. 8

[160] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1171–1180. ACM, 2017. 38

[161] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates. Fa* ir: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578. ACM, 2017. 2, 39, 40, 46, 52, 53, 57, 76, 77, 106, 108, 109, 112

[162] M. Zehlike and C. Castillo. Reducing disparate exposure in ranking: A learning to rank approach. *arXiv preprint arXiv:1805.08716*, 2018. 39, 40, 109, 110

[163] J. Zerilli, A. Knott, J. Maclaurin, and C. Gavaghan. Transparency in algorithmic and human decision-making: Is there a double standard? *Philosophy & Technology*, pages 1–23, 2018. 116, 123

[164] L. Zhou. Obama's new college scorecard flips the focus of rankings. *The Atlantic*, 2015. 5