DEXER: Detecting and Explaining Biased Representation in Ranking

Yuval Moskovitch
Ben Gurion University of the Negev
Beersheba, Israel
yuvalmos@bgu.ac.il

Jinyang Li
University of Michigan
Ann Arbor, MI, USA
jinyli@umich.edu

H. V. Jagadish
University of Michigan
Ann Arbor, MI, USA
jag@umich.edu

ABSTRACT

With the growing use of ranking algorithms in real-life decision-making purposes, fairness in ranking has been recognized as an important issue. Recent works have studied different fairness measures in ranking, and many of them consider the representation of different "protected groups", in the top-k ranked items, for any reasonable k. Given the protected groups, confirming algorithmic fairness is a simple task. However, the groups' definitions may be unknown in advance. To this end, we present DEXER, a system for the detection of groups with biased representation in the top-k. DEXER utilizes the notion of Shapley values to provide the users with visual explanations for the cause of bias. We will demonstrate the usefulness of DEXER using real-life data.

CCS CONCEPTS

• **Information systems** → *Retrieval models and ranking*; *Information retrieval diversity*.

KEYWORDS

ranking fairness; representation bias; explanations

ACM Reference Format:

Yuval Moskovitch, Jinyang Li, and H. V. Jagadish. 2023. DEXER: Detecting and Explaining Biased Representation in Ranking. In *Companion of the 2023 International Conference on Management of Data (SIGMOD-Companion '23), June 18–23, 2023, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3555041.3589725

1 INTRODUCTION

Ranking algorithms are an integral component of data-driven systems that are widely used in many application domains such as establishing credit scores [3], school admission [13], and hiring [7]. With the increasing awareness of algorithmic fairness, recent works have presented measures for fairness in ranking [14]. These definitions typically consider the representation of different protected groups, in the top-k ranked items, for any reasonable k. The notion of algorithmic fairness was studied extensively for a broad class of models [9, 14]. Fairness measures typically refer to a given "protected group" in the data, which is defined based on the values of some sensitive attributes (e.g., gender, race, age, or combinations thereof), usually based on the societal history of discrimination. Analyzing the fairness measure of a system with respect to the given group is a simple task. However, "non-standard" protected groups cannot always be specified in advance, and such groups may be overlooked when examining the performance of a system.

For example, a model developed to assign grades to students (in place of exams that were canceled due to the COVID-19 pandemic) was shown to be biased against high-achieving students from poor school districts[1]. For instance, students from low-income families were predicted to fail the Spanish exam, even when they were native Spanish speakers. In this case, the model was discriminating against Hispanic students from poor school districts. A primary source of bias was the use of historical exam results of each school to predict student performance. However, using the school (identified by school ID) to define the protected group is not an intuitive choice, and so may not have been considered. Moreover, even if we consider the group of Hispanic students as a protected group, we may not find any fairness issues, since this subgroup (of students scored unfairly on their Spanish exam) is only a small fraction of all Hispanic students.

A line of works has studied the problem of automatically detecting "problematic" or biased subgroups in the data without the need to specify the protected attributes a priori [4, 5, 8, 12], but these works considered only classification models. In [11] the authors of [12] extend their framework to consider ranking as well. However, their definition of biased subgroups builds on the notion of divergence to measure performance differences among data subgroups rather than fairness measures for ranking from the literature. Particularly, they do not consider a range of k, a key property for fairness in ranking. Intuitively, accounting for a range of k's ensures that the ranking is fair for any *position* in the ranking.

In this paper, we present DEXER (for Detecting and EXplaining biasEd Representation in ranking). DEXER is designed to detect groups with biased representation in the top-k ranked items (i.e., treated unfairly by a ranking algorithm) while eliminating the need to pre-define protected groups. The definition of biased representation is based on the notion of *proportional representation*, a commonly used measure for fairness in ranking (see, e.g., [16]). Intuitively, the representation of each group in the top-k should be proportional to its size in the data.

Given a group with biased representation, an analyst may wish to understand the cause of the bias. To this end, DEXER harnesses the notion of Shapley values [15] to identify attributes that significantly affect the ranking of the detected group. The system visualizes the value distribution of such attributes to analyze the

SIGMOD-Companion '23, June 18–23, 2023, Seattle, WA, USA

ACM ISBN 978-1-4503-9507-6/23/06.
https://doi.org/10.1145/3555041.3589694

[1] https://www.nytimes.com/2020/09/08/opinion/international-baccalaureate-algorithm-grades.html

#	Gender	School	Address	Failures	Grade	Rank
1	F	MS	R	1	11	8
2	M	MS	R	1	15	3
3	M	GP	U	1	8	10
4	M	GP	U	2	4	16
5	M	MS	R	0	19	2
6	F	MS	U	1	4	15
7	F	GP	R	1	7	11
8	M	GP	R	1	6	13
9	F	MS	R	0	14	4
10	F	MS	R	2	7	12
11	M	MS	R	2	13	6
12	F	GP	U	0	20	1
13	F	GP	U	2	12	7
14	M	MS	U	1	13	5
15	F	GP	U	1	5	14
16	M	GP	U	0	9	9

Figure 1: Students' data. The Rank column depicts their ranking based on the grade and number of past failures. The top-5 ranked students are highlighted

difference between the detected group and top-k ranked tuples. Shapley values have been used to provide explanations for regression and classification models, and their adjustment to our context holds two main challenges. The first is that Dexer aims to explain a ranking algorithm's result (rather than a regression or classification model). Moreover, it gets only the result of the ranking algorithm as input. The second challenge is the need to provide an explanation for a group (of tuples), while Shapley values are typically used to explain the outcome of a single tuple.

We will demonstrate the usefulness of Dexer using real-life data. We will walk the audience through the process of analyzing the results of a ranking algorithm, identifying groups with biased representation in the top-k ranked items, and let them interactively examine the cause for bias of different groups.

2 TECHNICAL BACKGROUND

We next (informally) introduce the model underlying Dexer using a running example and refer the readers to [10] for more details.

Example 2.1. The Student Performance Data Set [6] contains information from two Portuguese secondary schools in the Alentejo region of Portugal, Gabriel Pereira (GP) and Mousinho da Silveira (MS). The data was collected during the 2005-2006 school year and it contains the performance of 1044 students in the Math and the Portuguese language exams, along with demographic, social, and school-related information. Figure 1 depicts a sample from the data with the attributes: gender, school, address (urban or rural), and failures (number of past class failures). The grade attribute is in a scale of $0-20$. Consider a student excellence program committee that wishes to select students for a scholarship based on their academic achievements. To this end, they use a ranking algorithm R to rank students by their grades. In the case of similar grades, students with fewer failures are ranked higher. The scholars' list is publicly announced and should be diverse and inclusive.

2.1 Data groups and fairness measure

Groups in the data are represented using *patterns*, a set of attributes with values assignment. We say that a tuple $t \in D$ *satisfies* a pattern p if for every attribute in p, the value of t is similar to its assignment in p. The *size* of a pattern p in the dataset D is then the number of tuples in D that satisfy p.

Example 2.2. Consider the dataset given in Figure 1. p = {School = GP}, is an example of a pattern. Tuples 3, 4, 7, 8, 12, 13, 15, and 16 satisfy p and thus the size of p in the data is 8.

Fairness definitions for ranking algorithms typically account for the order of tuples in the output (or the top-k tuples), i.e., it is not enough to have adequate representation in the top-k, every prefix of the output should be "fair" as well. Accounting for a range of k's ensures that the ranking is fair for any *position* in the ranking. A prominent class of definitions considers the representation of each group in the data as a baseline measure for the representation in the top-k (see, e.g. [16]). Intuitively, for each group g and each k, the number of occurrences of items from g in the top-k ranked items should be proportional to the size of g in the dataset.

Example 2.3. Consider again the dataset given in Figure 1 and the ranker whose result is presented in the Rank column. The total number of students from each school (MS and GP) is 8. The total dataset size is 16, thus a proportionate representation of each school in the top-5 items should be roughly $5 \cdot \frac{8}{16} \approx 2$.

2.2 Problem Formulation

Our goal is to detect groups with biased representation in the top-k ranked items for a given ranking algorithm R, dataset D, and a range of k's. Following the line of work on proportional representation, we consider the representation of a group in the dataset as a baseline measure for its representation at the top-k. We say that a group has a biased representation in the output of a ranking algorithm R, if its size in the top-k ranked items by R is not proportionate to its representation in the data for any k in a given range of possible k's.

If we consider all possible data subgroups, the number of such groups can be extremely large. To this end, we let the user define a threshold τ on the reported group's size. Moreover, we wish to avoid reporting "very specific" descriptions of groups and provide the user with a concise set of properties that characterize meaningful groups (in terms of their size) that have biased representation. To this end, we present the notion of *most general* patterns. We say that a pattern p is the most general pattern with biased representation if p is used to represent a group with biased representation, and $\forall p' \subsetneq p$, the representation of p' in the top-k items is proportionate to its size in the dataset.

The problem is to report all meaningful (i.e., represented by most general patterns) substantial groups (i.e., large enough) with biased representation in the top-k ranked items for a given range of k. We let the user define the desired proportion of representation with respect to the size of the patterns in the data using a parameter $\alpha \in \mathbb{R}$. More formally, given a database D, ranked by a ranking algorithm R, a size threshold τ_s, a range $[k_{min}, k_{max}]$ and $\alpha \in \mathbb{R}$, our goal is to find for each $k_{min} \leqslant k \leqslant k_{max}$, all most general patterns p with size $\geqslant \tau_s$ such that $s_{R^k(D)}(p) < \alpha \cdot s_D(p) \frac{k}{|D|}$, where $s_{R^k(D)}(p)$ is the size of p in top-k ranked items in the dataset according to R and $s_D(p)$ is the size of p in D. Note that the ranking algorithm is treated as a black box, making the problem to be model agnostic.

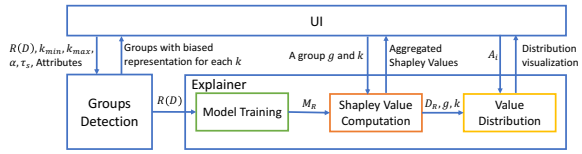


Figure 2: System architecture

3 SYSTEM OVERVIEW

Dexer's back-end is implemented in Python 3. The user interacts with the system using a dedicated user interface (shown in Figure 3), implemented in JavaScript using Retool[2]. The general architecture of the system is shown in Figure 2. We next briefly explain the components of the system.

3.1 Groups Detection

Dexer implements the algorithm for detecting groups with biased representation presented in [10]. We assume the data is represented using a single relational database, and that the relation's attribute values used for group definitions are categorical. To include attribute values drawn from a continuous domain in the group definition, we render them categorical by bucketizing them into ranges: very commonly done in practice to present aggregate results.

The algorithm is given a dataset D ranked by R (denoted by $R(D)$), a range of k (k_{min}, k_{max}), a size threshold τ_s and $\alpha \in \mathbb{R}$. To traverse the set of all possible groups (patterns), the algorithm uses the notion of pattern graph presented in [2]. Briefly, the nodes in the graph are the set of all possible patterns, and there is an edge between a pair of patterns p and p' if $p \subset p'$ and p' can be obtained from p by adding a single attribute with a value assignment. As shown in [2], the pattern graph can be traversed in a top-down fashion while generating each pattern at most once.

A simple solution for the problem is to traverse the different groups in the data and report those with biased representation in the top-k items for each k in a given range. However, note that the set of top-k and top-$(k+1)$ tuples differ by a single tuple. As a result, the search spaces for succeeding k values are typically very similar. The algorithm leverages this property and utilizes it to reduce the search space. It starts by performing a top-down search over the graph to find groups with biased representation in the top-k_{min} ranked items. In this phase, the algorithm utilizes the size threshold τ_s to prune the search space. After the first top-down search for k_{min} is done, the algorithm performs the search for every k from $k_{min}+1$ to k_{max}, where the starting point of the search for k is the endpoint of the search for $k-1$. The algorithm maintains a data structure to keep track of the patterns reached at the end of the search for each step and uses it to determine what parts of the pattern graph should be further explored when k is increased by 1, and a new tuple is introduced to the top-k. We experimentally evaluated our algorithm [10] and show that it runs much faster than the baseline solution (traversing all groups for each k), particularly as the number of attributes increases and the baseline becomes exponentially more expensive. Moreover, we show that in practical cases, the algorithm terminates within interactive time.

[2]https://retool.com/

3.2 Explainer

Once the groups with biased representation are detected, Dexer allows the user to explore the cause of the bias for different groups. This is done using the notion of Shapley values [15]. Shapley value is a concept adopted from game theory to explain the effect of different attributes on the output of a model for a given input. Given a regression model (or a classifier with probabilities) M, Shapley values are used to evaluate the contribution of each attribute on the output of M for a given input t.

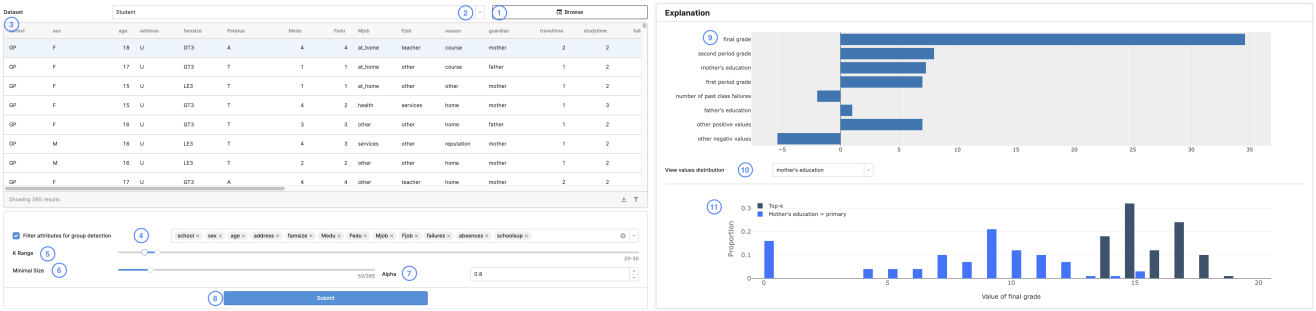
Intuitively, the cause for the bias is the values that affected the ranking of tuples in the given group. An explanation for the bias in the representation of a group consists of two parts. The first is a set of attributes with the highest effect on the ranking of tuples in the given group. The second is the values distribution of these attributes in the top-k and the biased represented group. There are two key challenges in adopting the notion of Shapley values to explain representation bias in ranking. First, Shapley values are typically used to explain the outcome of a regression model, whereas Dexer is given only the result of a ranking algorithm R (the ranking algorithm is a black box). Additionally, Shapley values are used to explain the contribution of the attribute values for a *single* tuple, while we are interested in explaining the (inadequate) representation of a group of tuples (in the top-k).

To address the first challenge, Dexer computes a regression model M_R that simulates the process of R and can be used to approximate the effect of attribute values of a given tuple t on t's ranking by computing the Shapley values of $M_R(t)$. To this end we define $D_R = \{(t, R(D)[t]) \mid t \in D\}$, where $R(D)[t]$ is the ranking of t in $R(D)$, and use it to train a regression model M_R. The training is done once the dataset is loaded and can be done using a separate background process while the user defines the attributes for group detection and explores the resulting groups. Then, to address the second challenge, we define the Shapley value of a group as the aggregated Shapley values for each tuple in the group. Given a pattern p such that p was returned by the algorithms for detecting groups with biased representation for a given k, we compute the Shapley values $(s_1^t, \ldots, s_m^t)$ for each tuple t such that t satisfies p, namely, for each tuple in the detected group. We then aggregate the results into a single Shapley value vector $(s_1, \ldots, s_m)$ for the pattern p such that $s_i = \frac{\sum_{t \text{ s.t. } t \text{ satisfies } p} s_i^t}{s_D(p)}$

Finally, to show the differences between the pattern p and the top-k patterns, we visualize the value distribution of attributes with large Shapley values of tuples that satisfy the pattern p compared to their distribution among the tuples in the top-k.

3.3 User Interface

The interaction with Dexer is done via a dedicated interface shown in Figure 3. In the input screen, shown in Figure 3a, the user can (1) upload a ranked dataset or (2) select a dataset from the preloaded datasets. Upon selection, the system (3) views the content of the selected datasets. By default, all the attributes are used for group detection, however, the user can select only a subset of them (4). Finally, the k range (k_{min}, k_{max}) (5), size threshold (τ_s)(6) and α values (7) can be set. When clicking on submit button (8) Dexer computes and presents the user the detected groups for each k with information about their size in the data and in the top-k (omitted



(a) Input screen **(b) Explanations view**

Figure 3: UI of Dexer.

from the presentation for space constraint). To get an explanation, the user can click on a group. Figure 3b shows an example for an explanation. It consists of the attributes with the highest aggregated Shapley values (9) of the selected group. The user can (10) select each one of the attributes and (11) view the value distribution of the selected attribute in the top-k and the selected group.

4 DEMONSTRATION PLAN

We will demonstrate the usefulness of Dexer in analyzing the fairness of ranking algorithms, detecting groups with biased representation, and the benefits of the explanations provided by the system using real-life datasets. In particular, we will use

- **The COMPAS Dataset**[3] was collected and published by ProPublica as part of their investigation into racial bias in criminal risk assessment software. It contains the demographics, recidivism scores produced by the COMPAS software, and criminal offense information. The data is ranked using the ranking method presented in [1].
- **Student Performance Dataset (Student dataset)**[4] shows the performance of students in secondary education of two Portuguese schools as described in Example 2.1. The data is ranked based on the value of the attribute G3 showing the student's math final grades.
- **German Credit Dataset**[5] with financial and demographic information. The data is ranked using the ranking presented in [16] based on creditworthiness.

We will walk the audience through the process of analyzing the results of a ranking algorithm. First, participants will be asked to select one of the pre-loaded datasets. We will browse through the selected dataset and invite the audience to choose attributes for groups detection, set k range (k_{min}, k_{max}), group size threshold τ_s, and α. We will let the audience explore the detected groups and then ask them to vary the input and observe the effect on the results.

Given the output of the first part, in the second part of the demonstration, we will let the audience interactively examine the cause for bias for the detected groups. The participants will be asked to select one of the groups detected by the algorithm and view the Shapley values computed by Dexer. Finally, by selecting an attribute from the list of attributes with the highest Shapley values, the system will visualize the distributions of the values of the attribute for tuples in the top-k, and the tuples in the group detected with biased representation, and the participants will be able to see the differences in the distributions.

Acknowledgments. This research was supported in part by NSF under grants 1741022, 1934565 and 2106176 and by ISF grant 2121/22.

[3]https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis
[4]https://archive.ics.uci.edu/ml/datasets/student+performance
[5]https://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

REFERENCES

[1] Abolfazl Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing fair ranking schemes. In *Proceedings of the 2019 International Conference on Management of Data*.
[2] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. 2019. Assessing and Remedying Coverage for a Given Dataset. In *ICDE*.
[3] Tobias Berg, Valentin Burg, Ana Gombović, and Manju Puri. 2020. On the rise of fintechs: Credit scoring using digital footprints. *The Review of Financial Studies* 33, 7 (2020).
[4] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. 2019. Fairvis: Visual analytics for discovering intersectional bias in machine learning. In *VAST*.
[5] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. 2020. Automated Data Slicing for Model Validation: A Big Data - AI Integration Approach. *IEEE Trans. Knowl. Data Eng.* 32, 12 (2020).
[6] Paulo Cortez and Alice Maria Gonçalves Silva. 2008. Using data mining to predict secondary school student performance. (2008).
[7] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. 2019. Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search. In *SIGKDD*. ACM.
[8] Zhongjun Jin, Mengjing Xu, Chenkai Sun, Abolfazl Asudeh, and HV Jagadish. 2020. Mithracoverage: a system for investigating population bias for intersectional fairness. In *SIGMOD*.
[9] Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. 2021. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application* 8 (2021).
[10] Yuval Moskovitch, Jinyang Li, and H. V. Jagadish. 2023. Detection of Groups with Biased Representation in Ranking. *CoRR* abs/2301.00719 (2023). https://arxiv.org/abs/2301.00719
[11] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. Identifying Biased Subgroups in Ranking and Classification. *CoRR* abs/2108.07450 (2021). https://arxiv.org/abs/2108.07450
[12] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. Looking for Trouble: Analyzing Classifier Behavior via Pattern Divergence. In *SIGMOD*.
[13] Christopher Peskun, Allan Detsky, and Maureen Shandling. 2007. Effectiveness of medical school admissions criteria in predicting residency ranking four years later. *Medical education* 41, 1 (2007).
[14] Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. 2022. Fairness in rankings and recommendations: an overview. *VLDB J.* 31, 3 (2022).
[15] L Shapley. 2020. 7. A Value for n-Person Games. Contributions to the Theory of Games II (1953) 307-317. In *Classics in Game Theory*. Princeton University Press.
[16] Ke Yang and Julia Stoyanovich. 2017. Measuring Fairness in Ranked Outputs. In *SSDBM*. ACM.