FaiRank: An Interactive System to Explore Fairness of Ranking in Online Job Marketplaces

Ahmad Ghizzawi[1], Julien Marinescu[2], Shady Elbassuoni[1], Sihem Amer-Yahia[2], Gilles Bisson[2]
[1]The American University of Beirut (Lebanon), [2]Univ. Grenoble Alpes, CNRS, LIG (France)
[1]{ahg05,se58}@aub.edu.lb, [2]firstname.lastname@univ-grenoble-alpes.fr

ABSTRACT

We demonstrate FaiRank, an interactive system to explore fairness of ranking in online job marketplaces. FaiRank takes as input a set of individuals and their attributes, some of which are *protected*, and a scoring function, through which those individuals are ranked for jobs. It finds a partitioning of individuals on their protected attributes over which fairness of the scoring function is quantified. FaiRank has several appealing features: (1) It can be used by different users: *the auditor* whose role is to monitor the fairness of ranking in a job marketplace, *the job owner* seeking to examine the influence of a scoring function and its variants on the ranking of candidates for a job, and *the end-user* who wants to assess the fairness of jobs on different marketplaces; (2) It is able to quantify fairness under different data and process transparency settings: when some attributes are anonymized and when only the ranking (and not the scoring function) is available; (3) It is interactive and lets its users explore different scoring functions and examine how fairness evolves; (4) It is generic and provides the ability to quantify different notions of fairness. Our demonstration will provide attendees with several scenarios for fairness of ranking in job marketplaces to experiment with and acquire an understanding of this important research question and its impact in practice.

1 INTRODUCTION

Freelancing marketplaces have become an online destination to find a temporary job. The ranking of individuals on platforms such as Qapa and MisterTemp' in France, and TaskRabbit and Fiverr in the USA, naturally poses the question of fairness. Fairness in ranking has recently received great attention from the data mining, information retrieval and machine learning communities (See for instance [1, 4, 6, 9, 10]). The most common definition of fairness in decision making was introduced in [2, 11] as *demographic parity*, and formalized in [3] as *group unfairness*. This definition captures the unequal treatment of a person based on *belonging to a certain group of people* defined using protected attributes such as gender and ethnicity. For instance, in the French Criminal Law (Article 225-1), 23 such attributes are listed as discriminatory.[1] The exact formulation of fairness varies and the purpose of FaiRank is to explore different formulations and unveil their impact on individuals.

User Roles. FaiRank appeals to different users. *The auditor*, whose role is to monitor the fairness of ranking in a marketplace, can use FaiRank to examine different jobs on that marketplace and quantify their fairness. *The job owner*, who wants to study the behavior of a scoring function and its variants, can use FaiRank to understand their impact on the ranking of individuals, and choose fairest one. Finally, *the end-user*, who is being ranked, can use FaiRank to assess the fairness of jobs on different marketplaces and make an informed decision.

Positioning. Most previous work on group-level fairness have either assumed that groups are pre-defined [9] or that they are defined using a single protected attribute (e.g., males vs females or whites vs blacks) [5]. FaiRank extends prior work to examine groups of people defined by *any combination of protected attributes* (the so-called *subgroup fairness* [6]). The scoring function yields one histogram per group as a score distribution. We use the Earth Mover's Distance (EMD) [8], a measure commonly used to compare histograms, to quantify the difference between score distributions across groups. The intuition is that if score distributions between groups differ significantly, the scoring function treats individuals in those groups unequally. This allows exploring different fairness formulations in FaiRank as any aggregation function over pairwise distances of score distributions in groups (highest average, lowest variance, etc.).

Since we do not want to focus only on pre-defined groups to quantify fairness, we must exhaust all possible ways of partitioning individuals into groups based on their protected attributes. This would capture cases where a scoring function treats males and females equally but is unfair to older African Americans compared to younger White Americans for instance. To examine all groups under different fairness definitions, we formulate an optimization problem as finding a partitioning of the ranking space, i.e., individuals and their scores, that exhibits some aggregation over pairwise partitions (e.g., the highest average EMD between partitions, the lowest average, the highest variance, etc.). Exhaustively enumerating all groups is exponential in the number of values of protected attributes. Therefore, to enable interactive response time, FaiRank relies on an efficient heuristic algorithm. At each step, the algorithm greedily splits individuals using the *most unfair* attribute according to the fairness definition. This local condition is akin to the one made in decision trees using gain functions [7]. The algorithm stops when there are no further attributes left to split on or when the current partitioning of individuals exhibits more unfairness than it would if its partitions were split further.

Data and Function Transparencies. In practice, data about individuals, i.e., their attributes, or the scoring function itself, may not be available. We integrate FaiRank with the k-anonymization ARX tool[2] and explore fairness for anonymized datasets. When the function is not available, FaiRank builds histograms using ranks of individuals rather than actual function scores.

Demonstration. Our demonstration combines the features of FaiRank to help attendees explore fairness of ranking in online job marketplaces and its impact in practice. It also sheds light

[1]https://www.legifrance.gouv.fr/affichCodeArticle.do?cidTexte=LEGITEXT000006070719&idArticle=LEGIARTI000006417828

[2]https://arx.deidentifier.org/

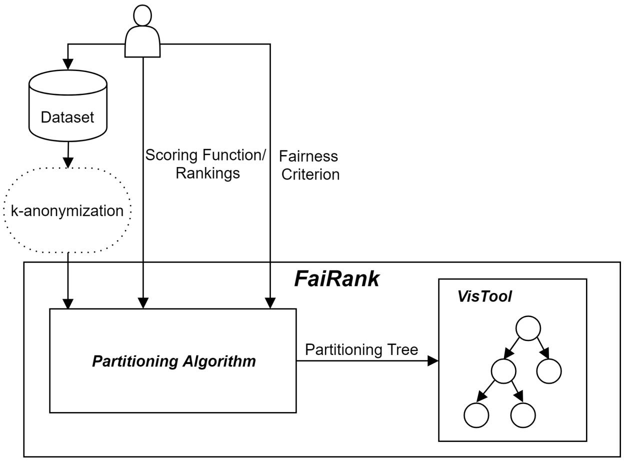

Figure 1: System architecture of FaiRank

on the interplay between data and function transparencies and the ability to quantify fairness. Additionally, FaiRank enables the exploration of different scoring functions, which can help choose the fairest for a given job. Finally, FaiRank can be used with standalone datasets and scoring functions, and since it can operate under various transparency settings, it can be used as a service to quantify fairness in existing blackbox job marketplaces.

2 SYSTEM OVERVIEW

Figure 1 depicts the system architecture of FaiRank. The user can select or upload a dataset which consists of a set of individuals and their attributes. The attributes can be protected such as gender, age, location, ethnicity, etc. or reflective of the performance or skills of the individuals such as reputation, knowledge in plumbing, writing skills, and mathematical abilities. Some of these attributes can also be anonymized. The user of the system can define or select a scoring function to rank individuals. The scoring function can be defined on a subset of the attributes of the individuals, for example a linear combination of an individual's reputation and plumbing skills, or of English writing skills and expertise in computer science. In addition, the user can filter the individuals based on protected attributes. This can be helpful in scenarios where the user is only interested in ranking a subset of individuals that satisfy certain criteria, say only individuals who speak Arabic or who are located in New York city. Instead of a scoring function, the user can also provide some ranking for the individuals (i.e., in the case that the scoring function is not transparent).

FaiRank solves an optimization problem that finds a partitioning of individuals over their protected attributes for which unfairness is subjected to an aggregation function (maximized, minimized, etc). The partitioning is displayed in a panel for the user. The user can interact with the returned partitions, view statistics such as the number of individuals in each partition, as well as a histogram of the scores of the individuals in each partition. The user can also choose to modify the scoring function or the fairness formulation, and obtain several panels to explore how that impacts fairness quantification. In the next section, we explain how we partition workers and quantify fairness of a scoring function given a set of individuals.

3 QUANTIFYING FAIRNESS

3.1 Model

To quantify fairness, we model the problem as aggregating a distance between the score distributions of all possible partitions of individuals. Unlike previous work where partitions were defined or known a priori (e.g., [5]), in FaiRank we explore the space of all possible groups defined by a combination of values of the individuals' protected attributes. The goal becomes finding an unfair partitioning of individuals under the scoring function. This can be formulated in many ways. For instance, the worst-case formulation would correspond to finding the *highest* distance between partitions. We cast this goal as an optimization problem as follows.

Definition 1 (Most Unfair Partitioning Problem). *We are given a set of individuals W, where each individual is associated with a set of protected attributes $A = \{a_1, a_2, ..., a_n\}$ and observed attributes $B = \{b_1, b_2, \ldots, b_m\}$. The protected attributes are inherent properties of the individuals such as gender, age, ethnicity, origin, etc. The observed attributes represent the skills of individuals for jobs and could include, for instance, the reputation and writing skills of an individual. We are also given a scoring function $f : W \longrightarrow [0, 1]$, which is defined using observed attributes as follows: $f(w) = \sum_{i=1}^{m} \alpha_i b_i$, where α_i is a user-defined weight for observed attribute b_i. A weight of zero indicates that the corresponding attribute is not relevant for the user in ranking the individuals. Our goal is to* fully *partition the individuals in W into k* disjoint *partitions $P = \{p_1, p_2, \ldots, p_k\}$ based on their protected attributes in A using the following optimization objective:*

$$\begin{aligned} &\underset{P}{argmax} && unfairness(P, f) \\ &\text{subject to} && \forall i, j \; p_i \bigcap p_j = \phi \\ & && \bigcup_{i=1}^{k} p_i = W \end{aligned}$$

Another formulation, *Least Unfair Partitioning Problem*, would be to find the partitioning that results in the smallest unfairness (i.e., *argmin* instead of *argmax* in the formulation above).

We now define how to compute the amount of unfairness of a function f for a partitioning P, or $unfairness(P, f)$ in the above optimization problem.

Definition 2 (Average Pairwise Unfairness). *For a set of individuals W, a full-disjoint partitioning of the individuals $P = \{p_1, p_2, \ldots, p_k\}$ and a scoring function f, unfairness of f for the partitioning P is quantified as the average pairwise Earth Mover's Distance (EMD) between the distribution of scores in the different partitions of P, which is computed as follows:*

$$unfairness(P, f) = \underset{i,j}{avg} \quad EMD(h(p_i, f), h(p_j, f))$$

where $h(p_i, f)$ is a histogram of the scores of individuals in p_i.

Another possible formulation is to compute unfairness as the maximum pairwise EMD, which would correspond to finding the partitioning with the highest maximum EMD between any pair of partitions.

Example. Consider the example dataset shown in Table 1 consisting of 10 individuals on a crowdsourcing platform who are ranked for some job using a scoring function f. Figure 2 shows one possible partitioning of those 10 individuals, that results

Table 1: An example dataset consisting of 10 individuals and a scoring function using Language Test and Rating

Individual	Gender	Country	Year of Birth	Language	Ethnicity	Experience	Language Test	Rating	f(w)
w1	Female	India	2004	English	Indian	0	0.50	0.20	0.29
w2	Male	America	1976	English	White	14	0.89	0.92	0.911
w3	Male	India	1976	Indian	White	6	0.65	0.65	0.65
w4	Male	Other	1963	Other	Indian	18	0.64	0.76	0.724
w5	Female	India	1963	Indian	Indian	21	0.85	0.90	0.885
w6	Male	America	1995	English	African-American	2	0.42	0.20	0.266
w7	Female	America	1982	English	African-American	16	0.95	0.98	0.971
w8	Male	Other	2008	English	Other	0	0.30	0.15	0.195
w9	Male	Other	1992	English	White	2	0.32	0.25	0.271
w10	Female	America	2000	English	White	5	0.76	0.56	0.62

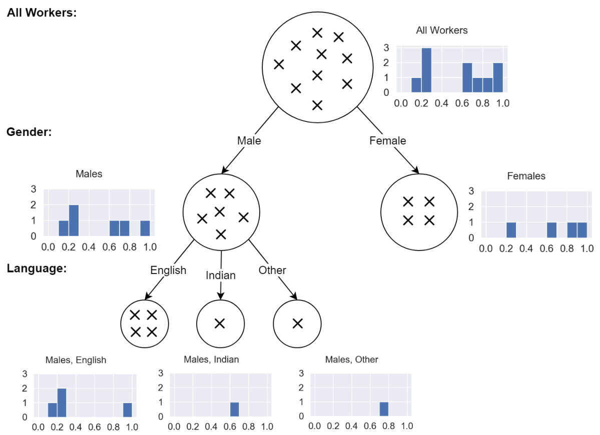

Figure 2: A partitioning of the example dataset

Algorithm 1 QUANTIFY(*current*: a partition, *siblings*: a set of partitions, f: a scoring function, A: a set of attributes)

```
if A = ∅ then
    Add current to output
else
    currentAvg = avg(EMD(current, siblings, f))
    a = mostUnfair(current, f, A)
    A = A − a
    children = split(current, a)
    childrenAvg = avg(EMD(children, siblings, f))
    if currentAvg ≥ childrenAvg then
        Add current to output
    else
        for each partition p ∈ children do
            QUANTIFY({p}, children − {p}, f, A)
        end for
    end if
end if
```

from splitting them based on Gender first, and then splitting only the Male partition based on Language to get the following partitioning of individuals: *Male - English, Male - Indian, Male - Other,* and *Female*. We quantify the unfairness of partitioning P as the average EMD between the pairs of partitions in P. To identify the most unfair partitioning, one must exhaust all possible *full disjoint* partitionings of individuals based on their protected attributes To do that, we generate a histogram for each partition as indicated in Figure 2 based on the function scores by creating equal bins over the range of f and counting the number of individuals whose function scores fall in each bin. The most unfair partitioning is then the one with maximum average pairwise EMD between its partitions' histograms.

3.2 Algorithm

Our optimization problem for finding the most unfair partitioning is hard since there are many possible partitionings P (exponential in the number of protected attribute values). For this reason, we propose an efficient heuristic algorithm to identify a partitioning of individuals with respect to our optimization objective within reasonable time. Our algorithm (pseudocode given as Algorithm 1) is *recursive*. We describe it with one unfairness formulation (the worst-case one provided in Equation 1 and with one aggregation function, namely average). Our algorithm decides whether or not to split a given partition by comparing the average EMD of that partition with its siblings to that of its children with its siblings. The intuition behind this is that it assesses what would happen to unfairness as measured by the average EMD if the partition was replaced by its children. It only splits a partition if its average pairwise EMD with its siblings is less than the average pairwise EMD of its potential children with the partition's siblings (that is in the case of the worst-case formulation of unfairness - other formulations require to change this test only). To invoke the algorithm for the first time, we first split the given set of individuals using the most unfair attribute and then the algorithm is called once for each resulting partition. After all recursive calls of the algorithm terminate, the output is returned as the final partitioning of the individuals.

4 DEMONSTRATION SCENARIOS

FaiRank caters to different user roles. A screenshot of the interface is shown in Figure 3. **A video of the demonstration is available at https://youtu.be/MckMJColcDk.** We propose to demonstrate it with 3 scenarios, one per role. During the demonstration, we will rely on two types of datasets, simulated datasets mimicking crowdsourcing platforms and real-data crawled from online freelancing marketplaces. In each case, we will explore various scoring functions representing different jobs as well as variants for the same job. We will also allow the exploration of transparency settings and their effect on fairness quantification, by making use of the ARX tool to k-anonymize the datasets[3] and by considering cases where the scoring function is available and cases where it is not.

[3] https://arx.deidentifier.org/

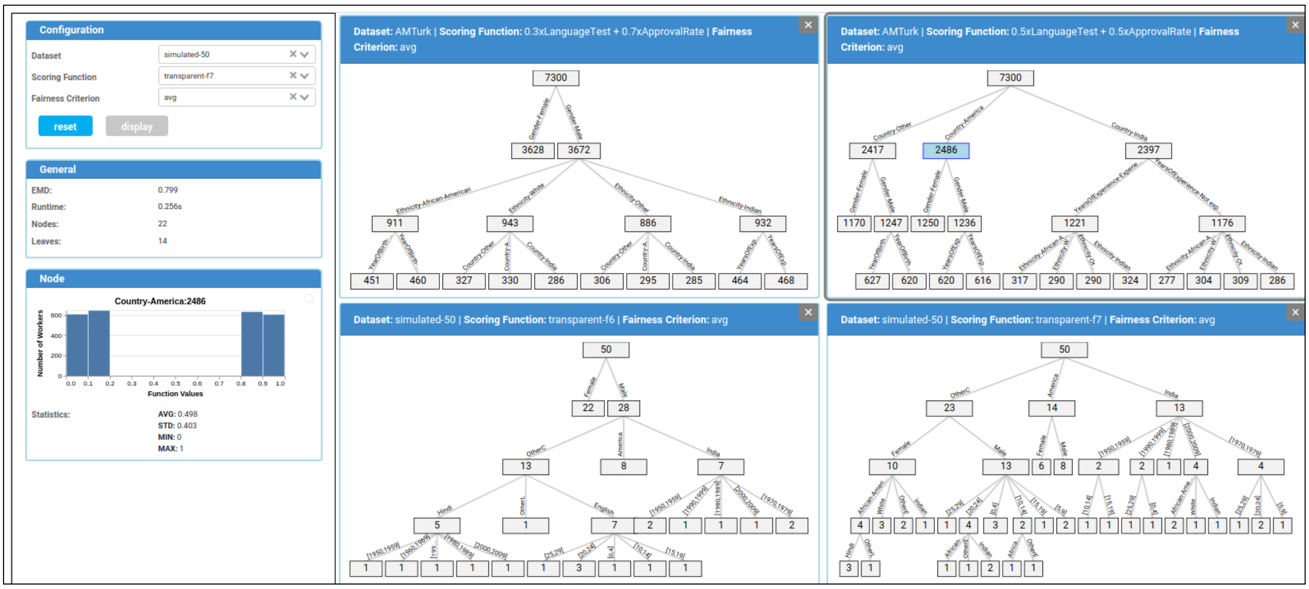

Figure 3: A snapshot of FaiRank's interface. The Configuration box on the left allows users to choose which dataset and which scoring functions they want to explore. It allows them to also choose a fairness criterion. The partitioning trees are displayed on the right in multiple panels, which allows the user to compare multiple scoring functions/datasets. General information about a partitioning tree can be found in the General box on the left, and the user can view statistics about a particular partition in the Node box by clicking on that partition in the tree.

AUDITOR Scenario. This scenario provides auditors with the ability to monitor a marketplace that offers multiple jobs, each with its own scoring function. It provides a big picture to auditors and lets them identify which jobs are most unfair to which individuals based on their rankings and under different notions of fairness. For instance, an auditor may be looking to draft a "fairness" report on a freelancing marketplace such as Qapa or TaskRabbit. The auditor would want to quantify the fairness for each job offered on the platform, and identify demographics groups that are least/most favored on the platform by each job. Additionally, the auditor might consider cases where the marketplace does not provide full transparency, either in terms of attributes of its users or in terms of the scoring functions used to rank those users, and we show the effect of this on quantifying fairness compared to the case when both attributes and the scoring function are available.

JOB OWNER Scenario. This scenario emphasizes the ability to define different scoring functions, and examine their impact on individuals. This exploration will help owners understand the behavior of their scoring functions and will guide them to choose the best function for their job, i.e., the one that satisfies some desired fairness. For instance, for an online job that requires people to write code, the owner can select those for whom the scoring function induces the least unfairness.

END-USER Scenario. This scenario offers end-users the ability to immerse themselves and simulate different cases in which they are to be ranked. For instance, an end-user wishing to find a job online, can examine how unfair some job is with respect to different groups of people. Given a group to which the end-user belongs (e.g., Young professionals in Grenoble) and a job of interest (e.g., installing wood panels), the end-user can see how well the marketplace is treating that group and make an informed decision of whether to target that job or not.

ACKNOWLEDGMENTS

This work is supported by the American University of Beirut Research Board (URB)

REFERENCES

[1] 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, SIGIR 2018*, pages 405–414, 2018.

[2] T. Calders and S. Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, Sep 2010.

[3] S. A. Friedler, C. Scheidegger, and S. Venkatasubramanian. On the (im)possibility of fairness. *CoRR*, abs/1609.07236, 2016.

[4] S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 2125–2126, 2016.

[5] A. Hannak, 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, CSCW 2017, Portland, OR, USA, February 25 - March 1, 2017*, pages 1914–1933, 2017.

[6] M. J. Kearns, S. Neel, A. Roth, and Z. S. Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 2569–2577, 2018.

[7] S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2(4):345–389, 1998.

[8] O. Pele and M. Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE, September 2009.

[9] A. Singh and T. Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 2219–2228, 2018.

[10] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 962–970, 2017.

[11] I. Zliobaite. A survey on measuring indirect discrimination in machine learning. *CoRR*, abs/1511.00148, 2015.