Exploring Fairness of Ranking in Online Job Marketplaces

Shady Elbassuoni [*1], Sihem Amer-Yahia [#2], Ahmad Ghizzawi [*1], Christine El Atie [*1]
[*]American University of Beirut, Lebanon, [#]CNRS, Univ. Grenoble Alpes, France,
[1] se58,@aub.edu.lb, cee11@mail.aub.edu, ahg05@mail.aub.edu,
[2] sihem.amer-yahia@univ-grenoble-alpes.fr

ABSTRACT

We study fairness of ranking in online job marketplaces. We focus on group fairness and aim to algorithmically explore how a scoring function, through which individuals are ranked for jobs, treats different demographic groups. Previous work on group-level fairness has focused on the case where groups are pre-defined or where they are defined using a single protected attribute (e.g., Caucasian vs Asian). In this paper, we argue for the need to examine fairness for groups of people defined with any combination of protected attributes. To do this, we formulate an optimization problem to find a partitioning of individuals on their protected attributes that exhibits the highest unfairness with respect to the scoring function. The scoring function yields one histogram of score distributions per partition and we rely on Earth Mover's Distance, a measure that is commonly used to compare histograms, to quantify unfairness. Since the number of ways to partition individuals is exponential in the number of their protected attribute values, we propose two heuristic algorithms to navigate the space of all possible partitionings to identify the one with the highest unfairness. We evaluate our algorithms using a simulation of a crowdsourcing platform and show that they can effectively quantify unfairness of various scoring functions.

INTRODUCTION AND POSITIONING

Online job marketplaces are gaining popularity as mediums to hire people to perform certain tasks. Examples include freelancing platforms such as Qapa and MisterTemp' in France, and TaskRabbit and Fiverr in the USA. On those platforms, workers can find temporary jobs in the physical world (e.g., looking for a plumber), or in the form of virtual "micro-gigs" such as "help with HTML, JavaScript, CSS, and JQuery". A person who needs to hire someone for a job can formulate a query and is shown a ranked list of people. The resulting ranking naturally poses the question of fairness. Algorithmic fairness has recently received great attention from the data mining, information retrieval and machine learning communities (See for instance [3, 5, 8, 10]). The most common definition of fairness was introduced in [1, 11] as *demographic parity*, that is the unfair treatment of a person based on *belonging to a certain group of people*. Groups are defined using protected attributes such as gender, age, ethnicity or location. We carry these definitions in our work and define unfairness in online marketplaces as the unequal treatment of people by a scoring function based on their protected attributes. This definition is inline with what is also commonly referred to as *group unfairness* [2].

Most previous work on group-level fairness have either assumed that groups are pre-defined [8] or that they are defined using a single protected attribute (e.g., Caucasian vs Asian) [4]. In this work, we consider groups of people defined with any combination of protected attributes (the so-called *subgroup fairness* [5]). The scoring function yields one histogram per demographic group as score distributions. We use the Earth Mover's Distance (EMD) [7], a measure that is commonly used to compare histograms, to quantify distances between groups. Our intuition is that if score distributions between groups are significantly different, the scoring function does not treat the individuals in these groups equally. For instance, consider two groups only, namely young males and females living in France. Unfairness can be computed as the distance between the score distributions of those two groups.

Since we do not want to focus only on pre-defined groups, we must exhaust all possible ways of partitioning individuals on their protected attributes to quantify unfairness. For example, a scoring function might treat both men and women equally but might be unfair towards older Asian Americans compared to younger White Americans. We define an optimization problem as finding a partitioning of the ranking space, i.e., individuals and their scores, that exhibits the highest average EMD between its partitions. Exhaustively enumerating all possible partitionings is exponential in the number of values of protected attributes. Therefore, we propose two heuristic algorithms, BALANCED that generates a balanced tree of partitions, and UNBALANCED that generates an unbalanced tree of partitions. At each step, our algorithms greedily split individuals on the worst attribute, i.e., the one that results in the partitioning with the highest EMD between score distributions. This local decision is akin to the one made in decision trees using gain functions [6]. The algorithms stop 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.

PROBLEM DEFINITION

To quantify unfairness in online job marketplaces, we model the problem as computing the highest average distance between the score distributions of all possible partitions of individuals. Unlike previous work where partitions were defined or known a priori (e.g., [4]), 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. 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 $B = \{b_1, b_2, \dots, b_m\}$. 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 $f : W \rightarrow [0, 1]$, which is defined using observed attributes*

Series ISSN: 2367-2005

10.5441/002/edbt.2019.77

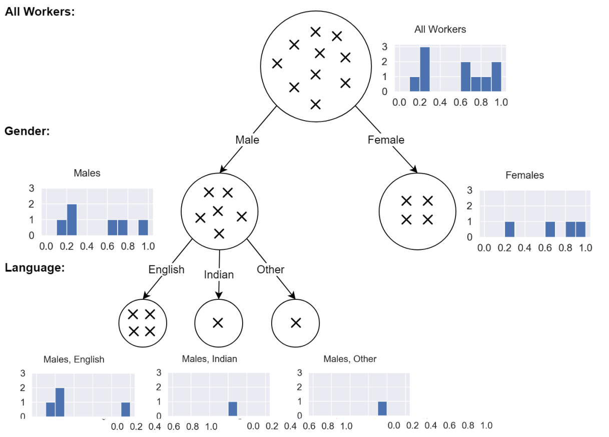

Figure 1: Optimum Partitioning of the Toy Example Data

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:*

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

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:*

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

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

Figure 1 shows a toy example of the optimum partitioning of 10 workers in a freelancing platform, which are ranked based on their qualification for some task using a scoring function f. The optimum partitioning is the one resulting from splitting the workers based on Gender first, and then splitting only the Male partition based on Language to get the following partitioning of workers: *Male - English, Male - Indian, Male - Other,* and *Female.* To arrive to this partitioning, one must exhaust all possible *full disjoint* partitionings of workers based on the protected attributes A and for each possible partitioning compute the average EMD between any two partitions. To do that, we generate a histogram for each partition as indicated in Figure 1 based on the function scores by creating equal bins over the range of f and counting the number of workers whose function values $f(w)$ fall in each bin.

Algorithm 1 BALANCED (W: a set of individuals, f: a scoring function, A: a set of attributes)

```
a = worstAttribute(W, f, A)
A = A − a
current = split(W, a)
currentAvg = averageEMD(current, f)
while A ≠ ∅ do
    a = worstAttribute(current, f, A)
    A = A − a
    children = split(current, a)
    childrenAvg = averageEMD(children, f)
    if currentAvg ≥ childrenAvg then
        break
    else
        current = children
        currentAvg = childrenAvg
    end if
end while
Add current to output
```

Once the partitioning with highest average pairwise unfairness has been identified, it is up to the user, requester or platform developer, to decide on the right subsequent action.

ALGORITHMS

Our optimization problem for finding the most unfair partitioning is hard since there are is an exponential number of possible partitionings P. For this reason, we propose heuristics-based algorithms to identify a partitioning of individuals with respect to our optimization objective within reasonable time.

We first propose BALANCED (Algorithm 1), an algorithm that generates a partitioning of the individuals in a greedy manner using the EMD of the partitions. BALANCED is based on decision trees with EMD as utility [6]. It starts by splitting the individuals on the *worst* attribute with respect to EMD. This is done by trying out all possible attributes one at a time, and associating to each attribute-value partition, one histogram of the scores of all the individuals it contains. For each candidate attribute, BALANCED computes the average pairwise EMD over histograms associated to the partitions obtained with the values of that attribute. It then returns the attribute with the highest average pairwise EMD and splits on that attribute. In the subsequent splitting steps, BALANCED iteratively partitions the individuals using the other attributes in the same manner and only stops when the average pairwise EMD of the current partitioning is greater than that of the next candidate partitioning.

BALANCED results in a partitioning of all the individuals using the same set of attributes (i.e., a balanced partitioning tree) since each splitting uses the same attribute over all current partitions. We also developed UNBALANCED (Algorithm 2), another algorithm that partitions the individuals in a non-homogenous manner by locally deciding for each partition whether to further split it or not (i.e., resulting in an unbalanced partitioning tree).

UNBALANCED is a *recursive* algorithm that decides 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

Algorithm 2 UNBALANCED (*current*: a partition, *siblings*: a set of partitions, f: a scoring function, A: a set of attributes)

```
if $A = \emptyset$ then
    Add $current$ to $output$
else
    $currentAvg = averageEMD(current, siblings, f)$
    $a = worstAttribute(current, f, A)$
    $A = A - a$
    $children = split(current, a)$
    $childrenAvg = averageEMD(children, siblings, f)$
    if $currentAvg \geq childrenAvg$ then
        Add $current$ to $output$
    else
        for each partition $p \in children$ do
            UNBALANCED $(\{p\}, children - \{p\}, f, A)$
        end for
    end if
end if
```

EMD of its potential children with the partition's siblings. To invoke the algorithm, we first split the given set of individuals using the worst attribute as in the case of BALANCED and then the algorithm UNBALANCED 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.

EVALUATION

To evaluate the effectiveness of our approach in quantifying unfairness, we run a simulation of a crowdsourcing platform using two sets of *active* workers and various scoring functions that rank those workers based on their qualification for tasks.

Setting. We generate two sets of active workers $\mathcal{W}$ of different sizes: 500 and 7300 (the estimated number of Amazon Mechanical Turk workers who are active at any time [9]). Each w in $\mathcal{W}$ has 6 protected attributes: Gender = {Male, Female}, Country = {America, India, Other}, Year of Birth = [1950, 2009], Language = {English, Indian, Other}, Ethnicity = {White, African-American , Indian, Other}, and Years of Experience = [0,30], and two observed attributes: LanguageTest = [25,100] and ApprovalRate = [25,100].

The values of those attributes are populated randomly so as to avoid injecting any bias in the data ourselves. Moreover, we define 5 different task qualification functions of the form $f = \alpha b_1 + (1 - \alpha) b_2$, where b_1 = Language Test and b_2 = Approval Rate and $\alpha \in \{0, 0.3, 0.5, 0.7, 1\}$.

We compare our two proposed algorithms UNBALANCED and BALANCED to a set of baselines. The first two baselines, which we refer to as R-BALANCED and R-UNBALANCED, are copies of our two algorithms BALANCED and UNBALANCED that use a random attribute instead of the worst attribute to split the workers at each step. The third baseline, which we refer to as ALL-ATTRIBUTES, is an algorithm that splits the workers based on *all* their protected attributes resulting in a full partitioning. Note that we also implemented an exhaustive algorithm that solves our optimization problem exactly by generating all possible partitionings in a brute-force manner and then returning the one with the highest average EMD. However, this algorithm failed to terminate after running for two days with only 6 attributes as in our simulation, even when each attribute had only a maximum of 5 values.

Simulation Results. Our first observation from Tables 1 and 2 is that for both datasets, functions f_4 and f_5 exhibit the highest unfairness as measured by the average pairwise EMD for all the partitionings retrieved by all the algorithms. Recall that these two functions are the ones that rely on one observed attribute only (LanguageTest in case of f_4 and ApprovalRate in case of f_5). *This indicates that if the scoring function uses fewer observed attributes, the chance of unfairness increases. In our simulation, since the attribute values were generated at random, there is a higher chance that the function scores correlate with a single protected attribute.*

Second, we observe that our two algorithms UNBALANCED and BALANCED consistently outperform or do as good as all other baselines for all datasets and functions. For the case of 500 workers, the UNBALANCED outperforms all other algorithms for the last three functions f_3, f_4 and f_5. On the contrary, the BALANCED returns the partitioning with the highest average EMD in the case of f_1. In the case of f_2, both BALANCED and R-BALANCED return the highest average EMD over the partitionings they find. *This allows us to validate the stopping condition used in our algorithms.*

Finally, in the case of 7300 workers, all the algorithms behave similarly, with the BALANCED and ALL-ATTRIBUTES returning the partitionings with slightly higher average EMD compared to all other algorithms. Upon investigating the returned partitioning by the different algorithms, we observed that in most cases all the algorithms returned the full partitioning tree, i.e., using all protected attributes, which is the same as the partitioning returned by the ALL-ATTRIBUTES algorithm. *We conjecture that it is due to the random values of all attributes.*

In terms of efficiency, the BALANCED algorithm took the most time to terminate compared to all other algorithms. In addition, the larger the dataset, the more time it took for all algorithms to finish. This is very intuitive given that the larger the dataset, the larger the individual histograms and the more time it takes to compute the pairwise EMD between them. Moreover, the deeper the partitioning tree, the larger the number of histograms that need to be compared. Finally, our two algorithms, BALANCED and UNBALANCED incur additional time since at each splitting step, they need to examine all remaining attributes to determine the worst one (i.e., the one which might result in the highest average EMD). All these factors contributed to the increased time to execute the BALANCED compared to all others. It is worth noting nonetheless that BALANCED terminates in less than 1.6 hours in the worst case (for the case of 7300 workers).

Qualitative Results. In addition to our simulation where we used a set of *random* task qualification functions, we also ran our algorithms on the following set of carefully-constructed functions, which are *unfair* by design:

- f_6: this function discriminates against females by setting the task qualification of workers as follows: $f_6(w) > 0.8$ if w is male and $f_6(w) < 0.2$ if w is female.
- f_7: this function sets the qualification of workers in a biased manner based on their gender and nationality as follows: $f_7(w) > 0.8$ if w is male and American, $f_7(w) < 0.2$ if w is female and American, $0.5 < f_7(w) < 0.7$ if w is Indian, either male or female, $f_7(w) > 0.8$ if w is female with any other nationality, and $f_7(w) < 0.2$ if w is male with any other nationality.
- f_8 designed as follows: $f_8(w) > 0.8$ if w is female and American, $0.5 < f_8(w) < 0.8$ if w is female and Indian and $f_8(w) < 0.2$ if w is female with another nationality.

Table 1: Average EMD and runtime for 500 workers and random functions

Algorithm	Average EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
UNBALANCED	0.195	0.191	**0.179**	**0.247**	**0.257**	20.987	23.715	22.823	29.504	28.845
R-UNBALANCED	0.193	0.193	0.177	0.243	0.253	28.33	26.871	28.354	27.333	28.372
BALANCED	**0.196**	**0.194**	0.177	0.246	0.253	**311.17**	**323.16**	**326.68**	**330.61**	**327.22**
R-BALANCED	0.195	**0.194**	0.177	0.246	0.253	131.87	122.49	119.97	127.06	124.46
ALL-ATTRIBUTES	0.195	0.193	0.177	0.246	0.253	42.708	42.494	42.597	42.235	42.337

Table 2: Average EMD and runtime for 7300 workers and random functions

Algorithm	Average EMD					time (in secs)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
UNBALANCED	0.161	0.162	**0.151**	0.208	0.209	1169.224	1246.651	1205.963	1292.506	1245.037
R-UNBALANCED	0.162	**0.163**	**0.151**	0.208	0.209	1401.36	1391.541	1358.795	1290.977	1397.894
BALANCED	**0.163**	**0.163**	**0.151**	**0.210**	**0.211**	**5733.528**	**5745.611**	**5693.681**	**5840.131**	**5808.715**
R-BALANCED	**0.163**	**0.163**	0.122	**0.210**	**0.211**	3174.327	3240.727	2358.744	3115.123	3120.553
ALL-ATTRIBUTES	**0.163**	**0.163**	**0.151**	**0.210**	**0.211**	1453.626	1449.466	1450.712	469.839	1467.606

Table 3: Average EMD for 7300 workers & biased functions

Algorithm	Average EMD			
	f_6	f_7	f_8	f_9
UNBALANCED	0.040	0.164	**0.460**	0.317
R-UNBALANCED	0.399	0.362	0.322	0.350
BALANCED	**0.800**	**0.427**	**0.460**	**0.359**
R-BALANCED	0.496	0.368	0.330	0.301
ALL-ATTRIBUTES	0.420	0.368	0.337	**0.359**

- f_9 correlates with protected attributes ethnicity, language and year of birth similarly to previous ones.

As can be seen from Table 3, BALANCED retrieves the partitionings with the highest possible average EMD compared to all other algorithms. In addition, the resulting partitionings are the ones expected, i.e., using the attributes for which the functions were designed to correlate with. For example, for f_6, BALANCED partitions the workers on only gender for all datasets. Similarly, for f_7, it partitions the workers on both gender and country. We show only the results in the case of 7300 workers due to space limitation. Finally, we observe that *overall for all functions and algorithms, the average EMD is much higher compared to the functions used in our simulation experiment, which indicates that our optimization problem is indeed effective in capturing unfairness of the scoring functions as conjectured.* The only exception was for UNBALANCED in the case of f_6 and f_7, where the algorithm ended up splitting the workers further than it should because of the local nature of its stopping condition. In fact, since the function scores were generated at random within the specified range, various runs of the experiments resulted in different behavior, where in some cases, UNBALANCED performed as well as BALANCED.

SUMMARY AND FUTURE WORK

We set out to examine fairness of ranking in online job marketplaces. To do this, we defined an optimization problem to find a partitioning of the individuals based on their protected attributes that exhibits the highest unfairness by a given scoring function. We used Earth Mover's Distance between score distributions as a measure of unfairness. Unlike previous work, we did not assume a pre-defined partitioning of individuals and instead proposed two heuristic algorithms, BALANCED and UNBALANCED, that efficiently partition the individuals without exploring the full space of partitionings. Our immediate plan is to test our algorithms on real datasets from Qapa and TaskRabbit. We are also investigating other formulations and metrics for fairness instead of the Earth Mover's Distance. We are also studying ways of "repairing" bias in the context of ranking in online job marketplaces.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Toon Calders and Sicco Verwer. 2010. Three naive Bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery* 21, 2 (01 Sep 2010), 277–292.

[2] Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. 2016. On the (im)possibility of fairness. *CoRR* abs/1609.07236 (2016).

[3] Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. 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.* 2125–2126.

[4] Aniko Hannak, Claudia Wagner, David Garcia, Alan Mislove, Markus Strohmaier, and Christo Wilson. 2017. 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.* 1914–1933.

[5] Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. 2018. 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.* 2569–2577.

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

[7] Ofir Pele and Michael Werman. 2009. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision.* IEEE, 460–467.

[8] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of Exposure in Rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018.* 2219–2228.

[9] Neil Stewart, Christoph Ungemach, Adam JL Harris, Daniel M Bartels, Ben R Newell, Gabriele Paolacci, Jesse Chandler, et al. 2015. The average laboratory samples a population of 7,300 Amazon Mechanical Turk workers. *Judgment and Decision making* 10, 5 (2015), 479–491.

[10] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. 2017. 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.* 962–970.

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