

AMERICAN UNIVERSITY OF BEIRUT

Discrimination-aware Task Assignment in
Crowdsourcing

by
Christine Elie El Atie

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Computer Science
of the Faculty of Arts and Sciences
at the American University of Beirut

Beirut, Lebanon
August 2018

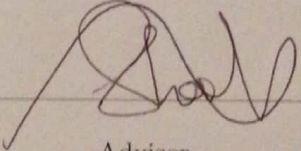
AMERICAN UNIVERSITY OF BEIRUT

Discrimination-aware Task Assignment in
Crowdsourcing

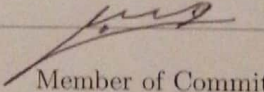
by
Christine Elie El Atie

Approved by:

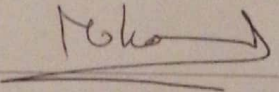
Dr. Shady Elbassuoni, Assistant Professor
Computer Science


Advisor

Dr. Wassim El Hajj, Chairperson and Associate Professor
Computer Science


Member of Committee

Dr. Mohamad Jaber, Assistant Professor
Computer Science


Member of Committee

Date of thesis defense: August 27, 2018

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: El Atie

Christine

Elie

Last

First

Middle

Master's Thesis

Master's Project

Doctoral Dissertation

I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One** ___ year from the date of submission of my thesis, dissertation or project.

Two ___ years from the date of submission of my thesis, dissertation or project.

Three ___ years from the date of submission of my thesis, dissertation or project.


Signature

September 13, 2018

Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

I would first like to thank my Thesis advisor Prof. Shady Elbassuoni for his constant guidance and help. Without him this work would not have been possible. I had no experience in research and he was extremely patient and taught me everything that I needed to know along the way.

Another thank you is due to Prof. Sihem Amer-Yahia from the "Laboratoire d'informatique de Grenoble" where I spent 3 months working on my thesis under her and Prof. Elbassuoni's guidance. She was of great help and had a huge impact on the direction this research took.

I would also like to thank my thesis committee members: Prof. Wassim El Hajj and Prof. Mohamad Jaber for accepting to serve on my committee and for their helpful suggestions towards improving my work.

An extra thank you also goes to Anas hosami for assisting in this work.

Finally, I must express my very profound gratitude to my parents, to my huge family and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

An Abstract of the Thesis of

Christine Elie El Atie for Master of Science
Major: Computer Science

Title: Discrimination-aware Task Assignment in Crowdsourcing

Algorithmic bias has been identified as a key challenge in many AI applications. One major source of bias is the data used to build these applications. For instance, many AI applications rely on crowdsourcing to generate training data. The generated data might be biased if the task assignment function is skewed towards certain groups of workers based on say gender, ethnicity or location. This typically happens as a result of a hidden association between the workers' qualifications for the task and the workers' attributes. Even in the case where such bias is intentional, e.g., in the case of positive discrimination, other biases may be hidden and can thus unintentionally favor acquiring data from certain groups of workers over others. In this thesis, we propose to quantify and address discrimination in crowdsourcing task assignment.

We define discrimination as the unbalanced targeting of workers by the task assignment function. To quantify discrimination, we formulate an optimization problem that partitions workers based on their attributes, computes the qualifications of workers in each partition, and finds the partitioning that exhibits the highest discrimination in task assignment decisions. Due to the combinatorial nature of our problem, we devise heuristics to navigate in the space of partitions. We also propose a way to address discrimination to achieve discrimination-free task assignment. Our experimental results on real and simulated data show that our approach can effectively unveil, quantify and address discrimination in crowdsourcing task assignment.

Contents

Acknowledgements	v
Abstract	vi
1 Introduction	1
1.1 Quantifying Discrimination.	4
1.2 Addressing Discrimination.	6
1.3 Empirical evaluation.	7
1.4 Thesis Plan	8
2 Literature Review	9
2.1 Algorithmic Discrimination.	9
2.2 Algorithmic Task Assignment.	10
2.3 Discrimination in crowdsourcing.	11
3 Setting	13
4 Approach	20

4.1	Quantifying Discrimination	20
4.2	Addressing Discrimination	25
5	Experiments	27
5.1	Quantifying discrimination	28
5.2	Evaluating the Algorithms	30
5.3	Addressing discrimination	34
6	Conclusion	40
	Bibliography	41

List of Figures

1.1	Distribution of the number of ratings for different worker groups in MovieLens.	5
3.1	A partitioning P_1 of workers in Table 3.1. The workers are partitioned based on language first then gender. The leaf nodes represent the final partitions in P_1	17
3.2	A partitioning P_2 of workers in Table 3.1. The workers are partitioned on country first, and then the workers from country = other only are further partitioned based on language. The leaf nodes represent the final partitions in P_2	18
4.1	Partitioning of workers in Table 3.1 using <code>BALANCED</code> with an obtained KL-divergence of 0.157.	22
4.2	Partitioning of workers in Table 3.1 using <code>UNBALANCED</code> with an obtained KL-divergence of 0.217.	24

List of Tables

3.1	Example Task-Assignment function for tweet annotation on 10 workers in a crowdsourcing platform.	14
3.2	Original ranking of workers L_O in Table 3.1 and updated rankings L_{P_1} and L_{P_2}	16
5.1	KL-divergence of the partitioning with maximum discrimination for the MovieLens dataset.	29
5.2	KL-divergence of the partitioning with maximum discrimination and the time taken to identify that partitioning on the simulated dataset using standardization.	32
5.3	KL-divergence and time taken to quantify the maximum discrimination on the simulated dataset using rescaling.	33
5.4	KL-divergence of the partitioning with maximum discrimination for the MovieLens dataset after score normalization using standardization.	35

5.5 KL-divergence of the partitioning with maximum discrimination
for the MovieLens dataset after score normalization using rescaling. 35

5.6 Mean and Standard deviation of the Euclidean distance of the
top-100 workers. 37

5.7 Mean and Standard deviation of the rating variance (f_1) and the
number of ratings (f_2) in D_2 for the top-100 workers. 38

Chapter 1

Introduction

A major source of algorithmic bias in AI is training data [1, 2, 3] and many AI applications rely on crowdsourcing to generate that data. In crowdsourcing, data is requested via Human Intelligence Tasks (HITs) and to ensure high-quality data, requesters rely on a task assignment function that utilizes the workers' qualifications for the task to assign tasks to workers . However, if the task assignment function is skewed towards certain groups of workers based on say gender, ethnicity or location, the model trained with that data is likely to be biased. Even if the task assignment function explicitly targets some groups of workers, e.g., in the case of positive discrimination [4], data may be biased with respect to subgroups within that group. This typically happens as a result of a hidden association between the workers' qualifications for task assignment and the workers' attributes. This association can unintentionally favor assigning tasks to certain groups of workers over others. The ability to detect such associations

is a necessary first step toward ensuring fairness in decision-making. In this thesis, we are interested in *unveiling, quantifying, and addressing* discrimination in crowdsourcing task assignment function.

While task assignment functions can filter out unqualified workers, they might also result in a skewed distribution of workers that are targeted by the task. For instance, it might be the case that while not intentional, the majority of workers that are allowed to attempt a task are males, white, young, or combinations of those. This is due to a hidden association between the task assignment function and the workers' attributes such as gender and age. Without identifying such associations, one runs the risk of acquiring biased data and training biased models. A recent incident that received a lot of attention in the media was regarding Google's image classifier, which has shown systematic bias in recognizing images of African American people [5]. This was mainly attributed to the fact that the software was not trained and tested by a diverse set of people. In general, without carefully characterizing the workers that are being targeted by a task assignment function, we always run the risk of acquiring biased data and making discriminatory decisions, which will definitely have a negative impact on the accuracy and generality of our application. We illustrate that in the following examples.

Example 1 (Tweet Sentiment Annotation) *Consider a crowdsourcing task that gathers training data for tweet sentiment annotation. The task consists of annotating a single tweet and some guidelines to help workers achieve the task.*

A typical task assignment function is the combination of acceptance ratio and language test. Naturally, only qualified workers will be selected. The resulting data may be skewed toward workers in some location and age group and may not cover enough views to train a robust model. In that case, we would like to quantify how discriminatory a task assignment function is with respect to worker groups, i.e., at which proportions does each worker group get assigned the annotation task and how different groups are targeted (e.g., young people in English-speaking countries will be more likely to participate than others).

Similarly, as the following example shows, we argue that discrimination in targeting workers may occur even in the case of positive discrimination, i.e., in the case where a specific group of workers is intentionally targeted for the task.

Example 2 (Positive Discrimination) *Consider a scenario that explicitly targets Europeans to gather diverse ratings on American blockbusters on a collaborative recommendation website. To achieve that, the task assignment function is applied to Europeans and computes their rating variance to target workers whose ratings are diverse. Such a function may be discriminatory with respect to different subgroups of Europeans: e.g., women whose rating variance is generally lower than men, or, French workers whose ratings are harsher than Spanish workers, etc. The ability to unveil that discrimination will shed light on the hidden associations between rating variance and worker groups, and help application developers make more informed decisions on whether this wanted positive discrimination is*

actually effective or if it is creating biases that are unaccounted for.

1.1 Quantifying Discrimination.

We define discrimination as the unbalanced targeting of workers by the task assignment function and advocate the need for an algorithmic approach for unveiling and quantifying it. Figure 1.1 is a scatter plot of the number of ratings for different groups of workers in MovieLens. As can be seen from the plot, the number of ratings vary across different groups, where some are under-represented and others are over-represented. For instance, older women tend to have fewer ratings compared to all other groups. This exhibits an inherent discrimination in the task assignment function with which these ratings were gathered. Note that if one were to examine gender only (i.e., number of ratings for males versus females) or age only (i.e., number of ratings for old, middle-aged, young and teen workers), it may seamlessly appear that the number of ratings across different groups are balanced. Examining combinations of these two attributes (gender and age) is what truly reveals discrimination. Our first goal in this thesis is thus to unveil and quantify discrimination induced by a task assignment function. To do so, we propose *to discover worker groups or partitions based on their attributes, in a data-driven fashion*. More precisely, for each possible partitioning of workers, we must examine the scores of workers using the task assignment function and quantify the difference in scores across partitions. Since there could be many

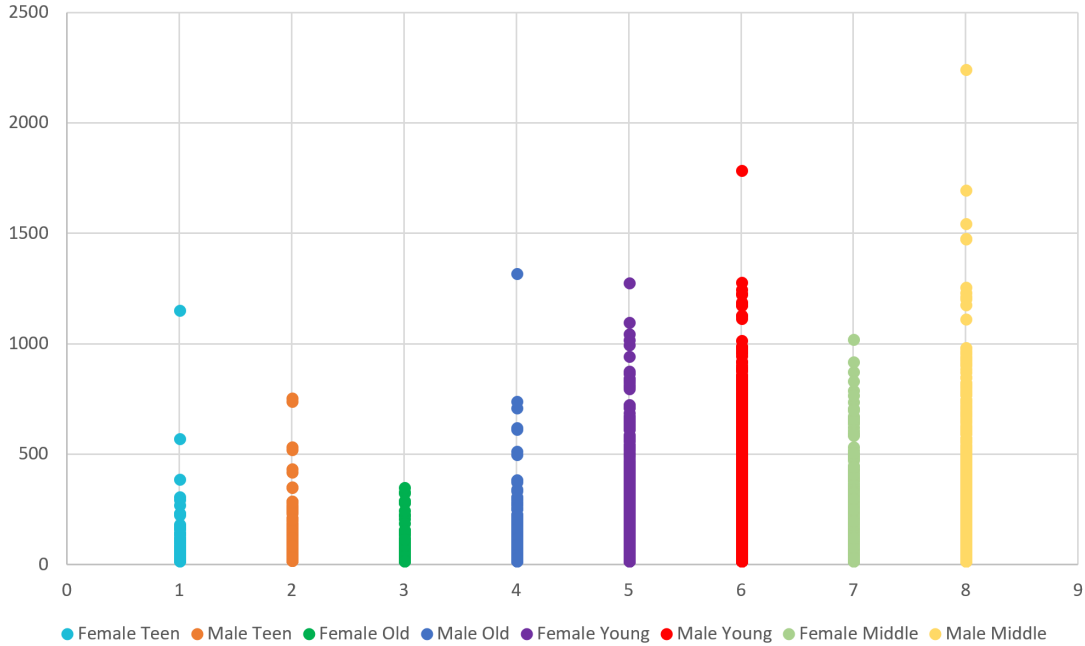


Figure 1.1: Distribution of the number of ratings for different worker groups in MovieLens.

possible ways of partitioning workers, and each way might result in a different amount of discrimination, we propose to model an optimization problem that finds a partitioning of workers, for which the task assignment function exhibits the *highest* discrimination. The rationale is that the partitioning with the highest discrimination will subsume all others.

To quantify discrimination of a worker partitioning, we compare the ranked list obtained by sorting workers on their function scores with a per-partition normalized ranking of workers. The intuition is that the higher the difference between the total ranking of the workers and the per-partition normalized ranking, the more different the distribution of scores is across the partitions. In this thesis, we use Kullback-Leibler divergence (KL-divergence) to compare two rank-

ings [6]. The highest discrimination of a task assignment function is the highest KL-divergence value we can obtain between a total ranking of workers and a per-partition normalized ranking. The problem of finding that value is naturally hard due to the combinatorial number of worker partitions. We propose to explore faster heuristics.

1.2 Addressing Discrimination.

Once discrimination is identified, we propose to address it. One possible way is to normalize the obtained scores across the identified partitions, i.e., those that exhibit the highest discrimination, to make them comparable. We refer to that approach as normalization-based. Consequently, one can then choose the K highest scoring workers after normalization or apply a threshold to filter out less qualified workers for instance. One may also argue that discrimination in task assignment can be addressed upfront by finding the most diverse set of workers based on their attributes and who are most qualified for the task. We refer to this approach as diversity-based and use a greedy algorithm to choose the K most qualified workers who are most diverse from each other [7].

1.3 Empirical evaluation.

Our evaluation aims to validate the usefulness of our approach on real datasets, validate our heuristics, and compare the normalization-based strategy for addressing discrimination to its baseline diversity-based strategy. In the first set of experiments, we show that our approach is successful in identifying the maximum discrimination of any given task assignment function on a MovieLens dataset. In the second set of experiments, we validate that our heuristics-based algorithms are more successful in identifying maximum discrimination on simulated data compared to baseline algorithms. Finally, in the third set of experiments, we demonstrate that the normalization-based strategy is superior to the diversity-based strategy as it achieves better representativity when acquiring ratings from workers in MovieLens, without sacrificing the quality of the acquired ratings. We also show that by addressing the identified discrimination using our normalization-based strategy, the resulting normalized task assignment function will be discrimination-free.

To summarize, we make the following contributions:

1. We define discrimination as the unbalanced targeting of workers by a task assignment function based on their attributes. We argue that unveiling and quantifying discrimination is necessary even in the case where some workers attributes are specified in the task assignment function.
2. We formalize discrimination quantification in a data-driven fashion as the

largest KL-divergence between a total ranking of workers and a per-partition normalized ranking. Due to the combinatorial space of partitions, we devise heuristics to compute discrimination in acceptable time.

3. We run several experiments. Our results show that our heuristics are fast without compromising discrimination values and that per-partition score normalization is necessary to acquire less-biased datasets.

1.4 Thesis Plan

The thesis is organized as follows: Chapter 1 begins by introducing this thesis. Chapter two then presents a detailed analysis of the previous work done in this field. Chapter three then describes our Setting and the context we will be working in. We then proceed to Chapter four that goes through the Approach we are taking to unveil, quantify, and address discrimination. To show that our approach is indeed useful we then present Chapter five that shows our experimental results using different datasets and many task assignment functions. Chapter 6 concludes this thesis.

Chapter 2

Literature Review

2.1 Algorithmic Discrimination.

Algorithms have replaced and outdone humans in many tasks but they often take biased decisions [8]. Discrimination was defined in [9] as the unfair treatment of a person based on belonging to a certain group of people rather than on individual merit. An example of discrimination was studied in [10] where a test was conducted to see if the name of a person, that directly relates to her skin color and gender, affects the ads shown when searching for her. It was shown that some specific names yielded discriminatory ads while others did not. Another example studied in [10], shows that a man surfing the Web gets ads for jobs that have higher income than the job ads that women get. A very close definition of discrimination is also presented in [11] as the act of favoring one group due to some attributes that might be included in the data-set or just inferred from the

context. Inferred attributes can lead to indirect discrimination. For example, ethnicity and postal code might be highly correlated and therefore removing ethnicity from the studied attributes will not have a great effect in avoiding discrimination [11]. To detect discrimination in algorithms, a framework [12] for "unwarranted associations" was designed to identify associations between a protected attribute, such as a person's race, and the algorithmic output using the FairTest tool. *In FairTest, these associations are typically assumed to be on a single-attribute level, which makes it different from our work where the goal is to quantify discrimination in treatment between worker partitions defined using a combination of multiple protected attributes.*

In our work, we see discrimination, as many before us, as favoring a person over another only due to her origin, gender, age, etc., which are attributes that are not necessarily specified in the task assignment function.

2.2 Algorithmic Task Assignment.

Several task assignment strategies exist in the literature, three of which were defined in [13] and are: 1) Relevance, where workers are assigned tasks that match their interests; 2) Diversity, where workers are tasks that match their interests and are diverse; 3) Div-Pay, where tasks are assigned to workers based on a combination of diversity and payment. In general, task assignment may or may not rely on workers' protected data. Some work shows that the use of

protected data is needed for a perfectly accurate outcome and proposes methods to balance fairness and accuracy [14]. To make an assignment fair, some accuracy may be sacrificed (e.g., balancing assignment between males and females may lead to less accurate assignments according to a ground truth). Other examples include adaptive methods, by assigning tasks adaptively in order to get more accurate and lower cost results when the set of workers available are diverse [15]. Similarly, some strategies assign tasks only to a few workers and infer the correct answers from them without having to assign the same task to many workers [16]. Some only focus on maximizing the requester’s gain to get the highest crowdwork quality at the lowest cost, by assigning tasks to the best matching workers (in terms of skills) [17]. Others tackle worker motivation [13] [18], worker problems such as the unfair rejection of work, or delayed or unfair payment [19].

2.3 Discrimination in crowdsourcing.

Several discrimination scenarios in task assignment were defined in [20]. That includes only accounting for requester preferences without quantifying how that affects workers, and vice versa. Another discriminatory scenario in [20] is related to worker’s compensation since a requester can reject work and not pay the worker or a worker can get under-paid. Discrimination in crowdsourcing can be defined for different processes. In this work, we focus on one process: task assignment.

In [21], the authors study ethics in crowd work in general. They analyze recent

crowdsourcing literature and extract ethical issues by following the PAPA (privacy, accuracy, property, accessibility of information) concept, a well-established approach in information systems. The review focuses on the individual perspective of crowd workers, which addresses their working conditions and benefits.

In [22], the notion of non-discrimination was defined as a measure between groups of people sharing the same value for some attributes. In that context, to assess discrimination mathematically, one needs to compare decisions between different groups of people that differ on the values of those attributes. This assumption states that all groups are equal in the construct space but there may be a structural bias in the observed space which leads to discrimination in decisions. Many techniques other than attribute grouping also exist, one of which was stated in [23] where they cluster points by separating them in a way to minimize the maximum intercluster distance. In our research, we adapt the definition in [22] for grouping workers on a combination of their protected attributes.

Chapter 3

Setting

We present our data model and define the problem of unveiling and quantifying discrimination in task assignment. We are given a set of workers W , a set of attributes $A = \{a_1, a_2, \dots, a_n\}$ and a set of qualifications $B = (b_1, b_2, \dots, b_m)$. Attributes in A are inherent properties of workers such as gender, age, ethnicity, origin, etc. Qualifications in B represent the abilities of a worker for completing a task. In crowdsourcing, qualifications include the acceptance ratio of the worker, language skills, mathematical abilities as measured by an analytical test and so on. On the social Web, a qualification may simply be the predicted rating of a worker for a movie or the opinion of a worker about a restaurant. Qualifications may be explicitly given by workers or inferred from previously rated items as in recommendation strategies [24], or from previously completed tasks in crowdsourcing [25].

A task-assignment function $f : W \rightarrow R$ calculates for a worker $w \in W$ a

Table 3.1: Example Task-Assignment function for tweet annotation on 10 workers in a crowdsourcing platform.

Worker	Gender	Country	YearOfBirth	Language	Ethnicity	Experience	LanguageTest	ApprovalRate	f(u)
W1	Female	America	2000	English	White	5	0.76	0.56	0.620
W2	Female	India	2004	English	Indian	0	0.50	0.20	0.290
W3	Male	America	1976	English	White	14	0.89	0.92	0.911
W4	Male	India	1976	Indian	White	6	0.65	0.65	0.650
W5	Male	Other	1963	Other	Indian	18	0.64	0.76	0.724
W6	Female	India	1963	Indian	Indian	21	0.85	0.90	0.885
W7	Male	America	1995	English	African-American	2	0.42	0.20	0.266
W8	Female	America	1982	English	African-American	16	0.95	0.98	0.971
W9	Male	Other	2008	English	Other	0	0.30	0.15	0.195
W10	Male	Other	1992	English	White	2	0.32	0.25	0.271

qualification score for the given task. For instance, for tweet annotation, f could simply be the location and language skill of the worker or a more sophisticated formula that aggregates the worker’s acceptance ratio on the platform, the quality of the worker’s past contributions, and the worker’s language skill. For a movie rating task, f could be the variance of ratings of the worker, or a sophisticated procedure such as a recommendation strategy that computes the expected rating of the worker for a movie.

The task assignment function f can make use of any attributes in A and qualifications in B . Its exact shape is not important for the purpose of our work. Our goal is to quantify the discrimination that happens as a result of applying f to workers in W for a given task in TA . Workers in W can be sorted in increasing or decreasing order of their scores computed by f . We refer to the resulting list as L_O . To quantify discrimination induced by f , we consider a full disjoint partitioning $P = \{p_1, p_2, \dots, p_k\}$ of the set of workers W on their attributes in A . Each worker must belong to *one and only one* partition p_i . Given a worker $w \in p_i$, We define $f'(w)$ as the normalized function score of worker w in partition

p_i . We experiment with two methods of normalization, namely standardization and rescaling. In the former, $f'(w)$ is computed as follows:

$$f'(w) = \frac{(f(w) - \mu_i)}{\sigma_i}$$

where

$$\mu_i = \frac{1}{|p_i|} \sum_{w \in p_i} f(w)$$

and

$$\sigma_i = \sqrt{\frac{1}{|p_i|} \sum_{w \in p_i} (f(w) - \mu_i)^2}$$

In the latter, $f'(w)$ is computed as follows:

$$f'(w) = \frac{f(w) - \min_i}{\max_i - \min_i}$$

where

$$\min_i = \min_{w \in p_i} f(w)$$

and

$$\max_i = \max_{w \in p_i} f(w)$$

We rank workers $w \in W$ based on their normalized function values $f'(w)$ to obtain a new ranking of workers L_P . Finally, we measure discrimination as $KL(L_P || L_O)$, which is the KL-divergence between the original ranking of workers L_O and the ranking of workers after per-partition normalization L_P . Intuitively,

the higher $KL(L_P||L_O)$ is, the more discrimination f induces on workers. The task assignment function f is said to not exhibit discrimination on workers W , if and only if there does not exist any full partitioning P of workers W such that $KL(L_P||L_O) \neq 0$.

Table 3.2: Original ranking of workers L_O in Table 3.1 and updated rankings L_{P_1} and L_{P_2} .

L_0	W8	W3	W6	W5	W4	W1	W2	W10	W7	W9	KL-divergence
L_{P_1}	W3	W8	W6	W5	W4	W1	W10	W7	W2	W9	0.020
L_{P_2}	W6	W8	W10	W3	W5	W4	W1	W2	W7	W9	0.081

Example. For example, Table 3.1 displays a set of workers W consisting of 10 workers, their attributes A (columns 2 to 7) and their qualifications B (columns 8 and 9). Assume that the task is tweet annotation and that f scores the workers $w \in W$ as follows:

$$f(w) = 0.3 \times \text{LanguageTest}(w) + 0.7 \times \text{ApprovalRate}(w)$$

The first row of Table 3.2 shows the original ranked list of workers L_O based on their f values. The second and third rows show the ranked lists of workers obtained from two different partitionings P_1 and P_2 , and the quantified discrimination of each partitioning as measured through KL-divergence between L_{P_1} and L_O , and L_{P_2} and L_O , respectively. The partitionings P_1 and P_2 are displayed in Figures 3.1 and 3.2.

Our discrimination quantification problem is hence the problem of finding a

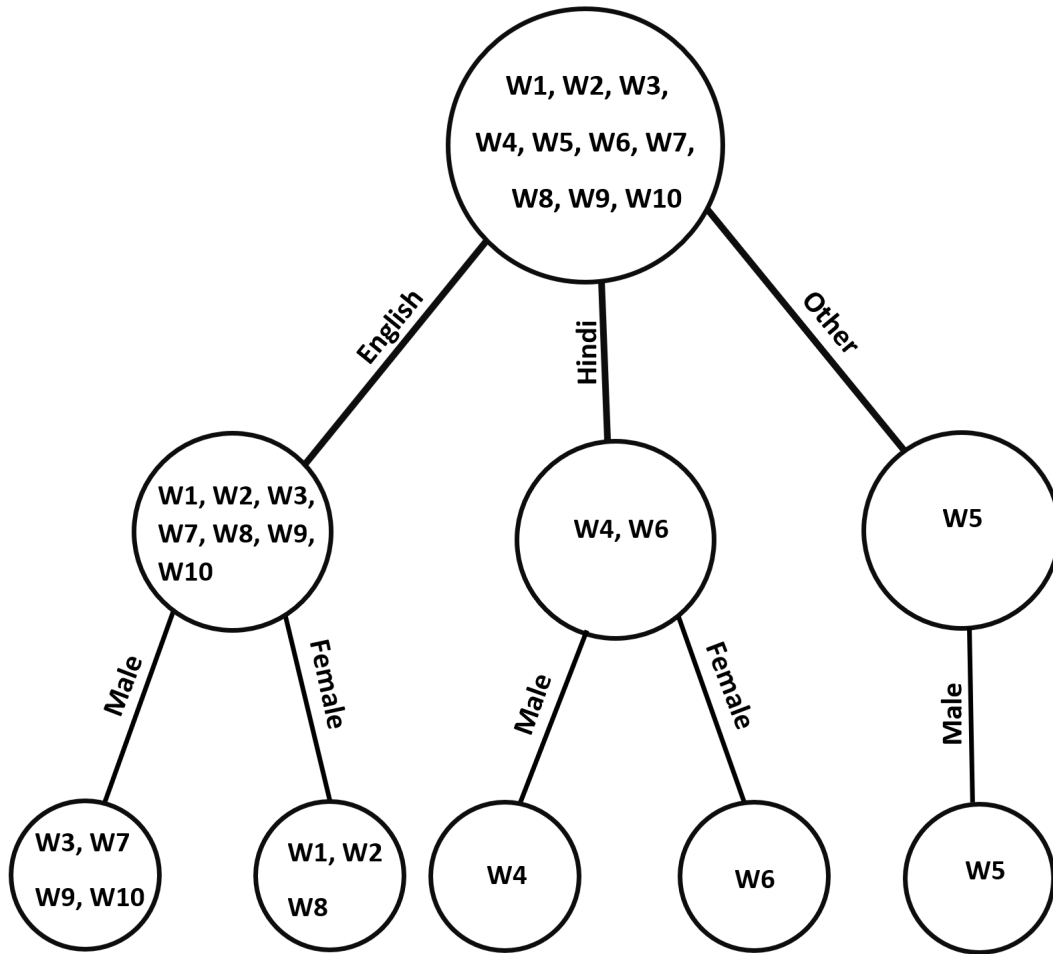


Figure 3.1: A partitioning P_1 of workers in Table 3.1. The workers are partitioned based on language first then gender. The leaf nodes represent the final partitions in P_1 .

full partitioning P of workers in W . One approach is to consider all possible partitionings of workers based on their attributes and retrieve the partitioning that returns the maximum discrimination as measured by the KL-divergence between the original ranking of workers L_O and the final ranking of workers L_P induced by per-partition normalization. *Intuitively, the partitioning with the maximum discrimination is the one that best captures the bias induced by the task*

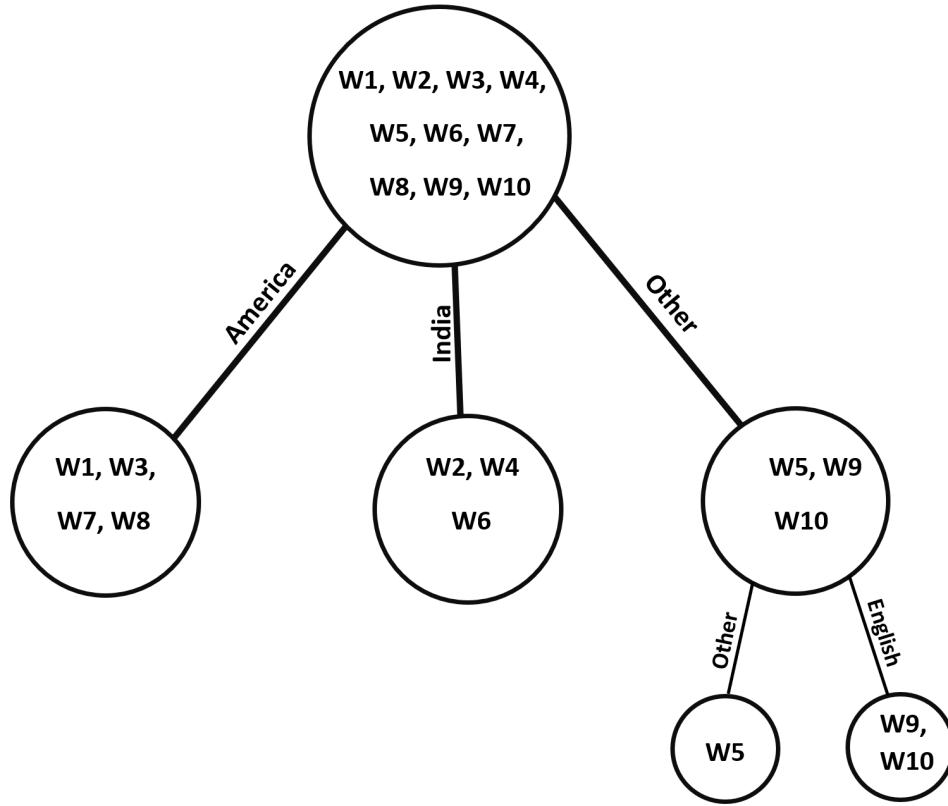


Figure 3.2: A partitioning P_2 of workers in Table 3.1. The workers are partitioned on country first, and then the workers from country = other only are further partitioned based on language. The leaf nodes represent the final partitions in P_2 .

assignment function f . Finding such a partitioning constitutes our optimization problem that we formulate as follows.

Definition 1 (Maximum Discrimination Partitioning Problem) : *Given a set of workers W and a task assignment function f , our goal is to fully partition workers in W into disjoint partitions $P = \{p_1, p_2, \dots, p_k\}$ based on their attributes*

in A using the following optimization objective:

$$\begin{aligned} \operatorname{argmax}_P \quad & KL(L_P||L_O) \\ \text{subject to} \quad & \forall i, j \ p_i \cap p_j = \phi \\ & \bigcup_{i=1}^k p_i = W \end{aligned}$$

where

$$KL(L_P||L_O) = \sum_{w \in W} r_P(w) \log \frac{r_P(w)}{r_O(w)}$$

where $r_O(w)$ is the rank of worker w in L_O and $r_P(w)$ is the rank of w in L_P .

It is obvious that our problem for finding the maximum discrimination partitioning is hard since there are many possible partitionings (exponential in the number of attribute values). For this reason, in the next chapter, we propose to develop heuristics-based algorithms to identify partitionings of workers with respect to our optimization objective within reasonable time. We will also describe how we propose to address discrimination once it is quantified by our algorithms.

Chapter 4

Approach

4.1 Quantifying Discrimination

As explained in the previous section, to quantify discrimination we rely on solving an optimization function that finds a partitioning of the workers that maximizes discrimination. Our optimization problem is hard due to the exponential number of possible partitionings. For this reason, we propose to use heuristics-based algorithms to identify partitionings of workers with high discrimination. We explore two such algorithms, which are greedy algorithms that rely on local decisions to maximize discrimination. Our algorithms rely on the same principle as decision partition trees that use a gain function to split a dataset [26]. In our case, the gain function relies on computing KL-divergence between rankings.

Our first algorithm `BALANCED` (Algorithm 1) takes as input a set of workers W , a task assignment function $f : W \rightarrow R$ and a set of attributes A for workers

in W . It returns a partitioning P of all workers in W . BALANCED starts by one partition containing all workers in W . It attempts to split that partition on the attribute that results in the highest KL-divergence between the normalized ranking of workers after the split and the current ranking. It then repeatedly tries to split workers on the remaining attributes and only stops when the KL-divergence between the current partitioning and the child’s is smaller than that of the current partitioning and the parent’s. Once it stops, it returns the obtained partitioning P , which is used to generate its final ranked list L_P .

Algorithm BALANCED makes use of two helper methods *normalize()* and *highestKLAttribute()*. *normalize()* takes a set of partitions (i.e., a partitioning) and a task assignment function, normalizes each partition using either standardization (mean and standard deviation) or rescaling (MIN-MAX) and returns a ranked list of the workers after their scores are normalized. *highestKLAttribute()* takes a set of partitions, a task assignment function and a set of attributes. It returns the attribute with the highest KL-divergence between the current ranking of workers based on the given partitions and the new ranking of workers after they are split using that attribute and performing a per-partition normalization of scores.

Figure 4.1 shows the partitioning P obtained using BALANCED and the KL-divergence between the original ranking L_O and the final ranking L_P for the dataset in Table 3.1. The workers are partitioned on their years of experience then on gender. The leaf nodes represent the final partitions in the obtained

partitioning.

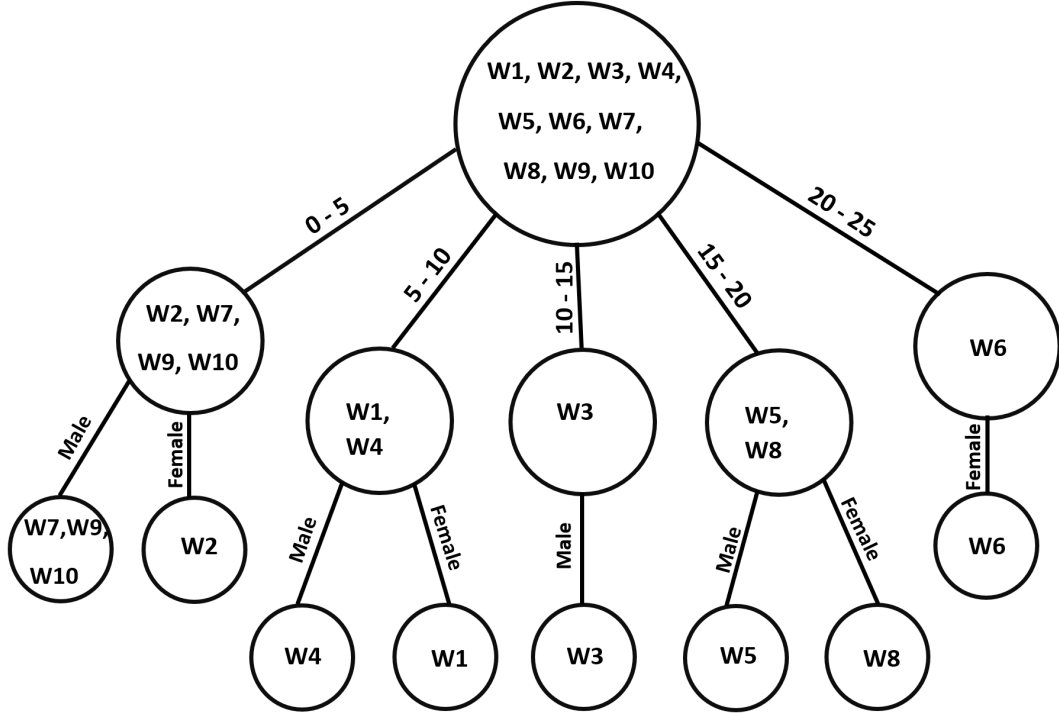


Figure 4.1: Partitioning of workers in Table 3.1 using BALANCED with an obtained KL-divergence of 0.157.

Algorithm BALANCED is an iterative algorithm that results in a balanced partition tree since the same attribute is used to split all current partitions. Our second algorithm UNBALANCED (Algorithm 2) produces an unbalanced tree by independently deciding for each partition whether to split it further or not. It takes as input two partitions, one representing a parent partition and the other representing a current partition for which a splitting decision is to be made. It also takes the siblings of the current partition, the task assignment function $f : W \rightarrow R$ and the set of worker attributes A . It then identifies for the current


```

1:  $P = \emptyset$ 
2:  $parent = W$ 
3:  $parentList = normalize(parent, f)$ 
4:  $a = highestKLAttribute(parent, f, A)$ 
5:  $A = A - a$ 
6:  $current = split(parent, a)$ 
7:  $currentList = normalize(current, f)$ 
8: while  $A \neq \emptyset$  do
9:    $a = highestKLAttribute(current, f, A)$ 
10:   $A = A - a$ 
11:   $child = split(current, a)$ 
12:   $childList = normalize(child, f)$ 
13:  if  $KL(currentList, parentList) \geq KL(childList, currentList)$  then
14:    break
15:  else
16:     $parent = current$ 
17:     $parentList = currentList$ 
18:     $current = child$ 
19:     $currentList = childList$ 
20:  end if
21: end while
22: add  $current$  to  $P$ 

```

Algorithm 1: BALANCED (U : a set of workers, f : a task assignment function, A : a set of attributes)

partition the attribute that would result in the highest KL-divergence between the current ranking of workers in that partition and the new ranking that would result from splitting workers using that attribute and normalizing their scores. It then splits the workers based on the identified attribute and then compares two KL-divergence values: one computed between workers in the current partition combined with its siblings and their parent partition, and one computed between workers in the children of the current partition combined with the current partition siblings and the parent partition. Algorithm UNBALANCED should be initially invoked by splitting the set of all workers W using the attribute that

would result in the highest KL-divergence and then calling the algorithm for each resulting partition. Again, once it terminates, it returns the final partitioning P which can then be used to generate the final ranked list of workers L_P .

Figure 4.2 shows the partitioning P obtained using UNBALANCED and the KL-divergence between the original ranking L_O and the final ranking L_P for the dataset shown in Table 3.1. The workers are partitioned on their years of experience and then only the partitions containing workers with years of experience between 0 and 5, and between 10 and 15 are further split on gender. The leaf nodes represent the final partitions in the obtained partitioning.

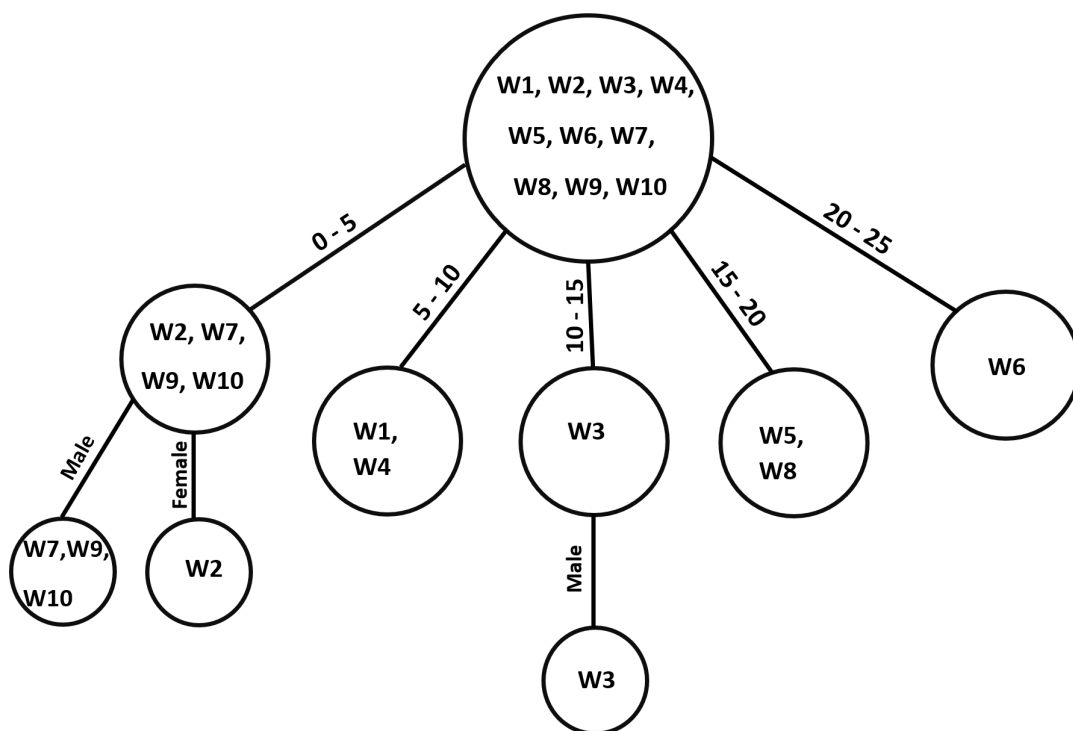


Figure 4.2: Partitioning of workers in Table 3.1 using UNBALANCED with an obtained KL-divergence of 0.217.

These two algorithms are quadratic as a maximum depending on the number

```

1: if  $A = \emptyset$  then
2:   add current to  $P$ 
3: else
4:    $parentList = normalize(parent, f)$ 
5:    $currentList = normalize(current \cup siblings, f)$ 
6:    $a = highestKLAttribute(current, f, A)$ 
7:    $A = A - a$ 
8:    $children = split(current, a)$ 
9:    $childrenList = normalize(children \cup sibling, f)$ 
10:  if  $KL(currentList, parentList) \geq KL(childrenList, parentList)$  then
11:    add current to  $P$ 
12:  else
13:    for each partition  $child \in children$  do
14:      UNBALANCED ( $current, child, children - \{child\}, f, A$ )
15:    end for
16:  end if
17: end if

```

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

of times they loop and the number of attributes since each time we’re checking all remaining attributes and then deciding whether to continue splitting or not.

4.2 Addressing Discrimination

Once a partitioning P is obtained, whether with BALANCED or with UNBALANCED, we propose to address discrimination as follows. First, we normalize the workers’ function scores for each partition $p \in P$ using one of the normalization techniques described in Section 3 to obtain new function scores $f'(w)$. We then re-rank all workers in all partitions globally based on their new scores $f'(w)$ and return the new ranked list L_P along with the new scores of workers $f'(w)$ obtained

after normalization.

Chapter 5

Experiments

We run three sets of experiments. In the first set, we apply `BALANCED` and `UNBALANCED` on a dataset generated from MovieLens using various task assignment functions and show that our algorithms are successful in quantifying discrimination. In the second set of experiments, we compare our proposed algorithms to a set of baselines to validate our heuristics in identifying maximum discrimination using simulated data. Finally, in the third set of experiments we show that by addressing the identified maximum discrimination using our normalization-based strategy, the resulting normalized task assignment function will exhibit less discrimination. We also compare our normalization-based strategy for addressing discrimination with a baseline diversity-based strategy.

5.1 Quantifying discrimination

We test BALANCED and UNBALANCED on the MovieLens 1M dataset ¹. The dataset consists of 6040 workers with around 1 million ratings in total (972599 to be exact). Each worker is associated with four attributes, namely gender (Male or Female), age (Teen, Young, Middle-aged or Old), occupation (one of 22 different occupations), and location (one of 50 states).

We examine three different task assignment functions. The first function aims to acquire ratings from workers who have a *diverse* set of ratings. To this end, we use rating variance as the task assignment function, i.e.,:

$$f_1(w) = \frac{1}{|I_w|} \sum_{i \in I_w} (r_w(i) - \mu_w)^2$$

where I_w is the set of movies that worker w rated, $r_w(i)$ is the rating provided by worker w for movie i , and μ_w is the average rating provided by the worker w for all the movies she rated. To discard workers with a very small number of ratings, our task assignment function only consider workers who have rated more than 100 movies. The second function aims to acquire ratings from the *least* active workers. To this end, we use a task assignment function f_2 to rank workers based on the number of movies they rated. Our third function does not exhibit any discrimination by construct. It utilizes a task assignment function f_3 that takes a worker and returns 0 if the worker has rated fewer than 20 movies

¹<https://grouplens.org/datasets/movielens/1m/>

and 1 otherwise. The reason we opted for a threshold of 20 is that MovieLens required at least 20 ratings per worker in the dataset.

Table 5.1: KL-divergence of the partitioning with maximum discrimination for the MovieLens dataset.

Algorithm	Standardization			Rescaling		
	f_1	f_2	f_3	f_1	f_2	f_3
UNBALANCED	0.082	0.080	0.512	0.153	0.122	0
BALANCED	0.074	0.065	0.499	0.103	0.103	0

Table 5.1 shows the KL-divergence of the partitioning that exhibits the maximum discrimination as measured by the KL-divergence between the original ranking of workers and the final ranking after score normalization, using both standardization (i.e, mean and standard deviation), and rescaling (i.e, MIN-MAX). In the case of rescaling (second half of the table), our algorithms unveiled some discrimination for the first two functions f_1 and f_2 and no discrimination in the case of the third function f_3 , as indicated by a KL-divergence of 0. On the other hand, using standardization as a normalization technique (first half of the table), our algorithms unveiled some discrimination for all three functions (i.e., KL-divergence > 0). This is due to the fact that standardization makes use of the mean and standard deviation of partitions, which highly depend on the size of partitions (number of workers). That is, if two partitions have different sizes but the same score distribution, the scores after normalization end up being different. *In general, when the workers are not evenly distributed with respect to the attributes, i.e., there are many more males than females or younger people than*

older ones, using rescaling normalization might be more effective in quantifying discrimination as it is less sensitive to the sizes of the partitions.

5.2 Evaluating the Algorithms

The goal of this set of experiments is to evaluate our heuristics-based algorithms, BALANCED and UNBALANCED, for quantifying maximum discrimination. To do this, we simulate a crowdsourcing platform consisting of 20,000 workers. We then sample three different datasets of *active* workers from the platform. The first dataset consists of 50 active workers (i.e., $|W| = 50$). The second consists of 500 active workers and the third consists of 7300 active workers, which is estimated to be the size of active workers on Amazon Mechanical Turk [27]. Each $w \in W$ has 6 attributes, as follows: 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 qualifications: language test $q_1 = [25,100]$ and approval rate $q_2 = [25,100]$. The values of the attributes and qualifications for each worker are set at random. We also generate five different task assignment functions $f_i(w)$ that score workers based on their qualifications as follows:

1. $f_1(u) = 0.3 * q_1 + 0.7 * q_2$

2. $f_2(u) = 0.7 * q_1 + 0.3 * q_2$

3. $f_3(u) = 0.5 * q_1 + 0.5 * q_2$

4. $f_4(u) = 1 * q_1 + 0 * q_2$

5. $f_5(u) = 0 * q_1 + 1 * q_2$

We compare BALANCED and UNBALANCED to three baselines. The first two baselines are copies of our two algorithms, which we refer to as R-BALANCED and R-UNBALANCED and which use a random attribute to split partitions rather than the attribute that would result in the maximum KL-divergence between the current partition(s) and their children. R-BALANCED and R-UNBALANCED are used to validate our splitting heuristic that greedily picks the attribute that results in the highest KL-divergence. The third baseline is an algorithm that partitions workers on all attributes, which we refer to as FULL and which is used to validate our stopping condition that is triggered when splitting the current partition(s) does not result in an increase in the KL-divergence. We know that the first baseline that comes to mind is the exhaustive approach, where we find the most discriminatory partitioning among all possible partitionings. We tried running this exhaustive algorithm on our dataset using the attributes that we have. The algorithm ran for a very extended period of time and needed a lot more even when it was very optimal and this is due to the large number of attributes and sub-attributes. That's why we decided to not rely on this baseline and use the three that we have here. Table 5.2 and 5.3 displays the KL-divergence between the original ranking of workers and the ranking after normalizing their

scores in each of the identified partitions using standardization and scaling.

Table 5.2: KL-divergence of the partitioning with maximum discrimination and the time taken to identify that partitioning on the simulated dataset using standardization.

Algorithm	KL-divergence					Time (in secs.)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
Results for 50 workers										
UNBALANCED	0.212	0.248	0.170	0.232	0.216	0.056	0.059	0.055	0.06	0.052
R-UNBALANCED	0.132	0.142	0.108	0.150	0.130	0.029	0.036	0.030	0.028	0.028
BALANCED	0.078	0.088	0.093	0.074	0.056	0.022	0.022	0.022	0.074	0.025
R-BALANCED	0.099	0.129	0.073	0.085	0.100	0.006	0.006	0.006	0.006	0.006
FULL	0.007	0.002	0.003	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Results for 500 workers										
UNBALANCED	0.174	0.194	0.175	0.193	0.190	0.668	0.668	0.682	0.686	0.657
R-UNBALANCED	0.130	0.126	0.132	0.134	0.116	0.433	0.389	0.398	0.401	0.367
BALANCED	0.131	0.123	0.129	0.122	0.128	0.326	0.318	0.319	0.322	0.321
R-BALANCED	0.076	0.082	0.093	0.064	0.107	0.077	0.067	0.089	0.068	0.084
FULL	0.039	0.034	0.036	0.028	0.032	0.012	0.013	0.012	0.011	0.033
Results for 7300 workers										
UNBALANCED	0.064	0.062	0.062	0.064	0.068	25.675	23.436	25.403	22.310	24.561
R-UNBALANCED	0.032	0.035	0.033	0.031	0.037	16.574	16.423	15.481	14.806	15.002
BALANCED	0.110	0.108	0.107	0.111	0.116	26.322	25.803	25.211	24.787	25.585
R-BALANCED	0.110	0.108	0.107	0.056	0.116	5.942	6.010	5.995	4.698	5.847
FULL	0.110	0.108	0.107	0.111	0.116	0.872	0.882	0.917	0.887	0.844

As can be seen from Table 5.2, in the majority of cases, BALANCED and UNBALANCED outperform all baselines by finding a higher KL-divergence. More precisely, UNBALANCED outperforms R-UNBALANCED for all datasets and for all task assignment functions using both normalization techniques. On the other hand, BALANCED performs worse than its random counterpart R-BALANCED in terms of KL-divergence only 4 times out of 15 when standardization is used as a normalization technique and only once when rescaling is used as a normalization strategy. Overall, despite making local decisions, our greedy approach to choose the attribute that yields the highest KL-divergence before and after splitting results in higher KL-divergence between the final ranking of workers and the

Table 5.3: KL-divergence and time taken to quantify the maximum discrimination on the simulated dataset using rescaling.

Algorithm	KL-divergence					Time (in secs.)				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
Results for 50 users										
UNBALANCED	0.260	0.336	0.303	0.292	0.280	0.026	0.028	0.025	0.026	0.029
R-UNBALANCED	0.212	0.211	0.213	0.213	0.176	0.013	0.013	0.013	0.014	0.014
BALANCED	0.150	0.167	0.134	0.075	0.131	0.011	0.010	0.011	0.015	0.011
R-BALANCED	0.136	0.137	0.181	0.106	0.104	0.003	0.003	0.002	0.003	0.003
FULL	0.007	0.005	0.009	0.006	0.009	0.001	0.001	0.001	0.001	0.001
Results for 500 users										
UNBALANCED	0.254	0.231	0.249	0.241	0.264	0.345	0.342	0.341	0.352	0.354
R-UNBALANCED	0.204	0.189	0.206	0.197	0.190	0.193	0.171	0.187	0.207	0.177
BALANCED	0.196	0.172	0.189	0.166	0.195	0.187	0.179	0.185	0.185	0.189
R-BALANCED	0.108	0.092	0.119	0.068	0.128	0.046	0.040	0.046	0.046	0.044
FULL	0.065	0.058	0.063	0.055	0.065	0.011	0.010	0.010	0.022	0.010
Results for 7300 users										
UNBALANCED	0.068	0.070	0.077	0.053	0.047	21.549	21.073	19.895	18.323	17.677
R-UNBALANCED	0.038	0.037	0.047	0.006	0.005	12.792	12.824	12.309	8.670	10.523
BALANCED	0.146	0.146	0.150	0.136	0.136	23.019	23.380	22.045	21.136	21.228
R-BALANCED	0.146	0.146	0.150	0.045	0.045	5.278	5.234	5.079	3.180	3.153
FULL	0.146	0.146	0.150	0.136	0.136	0.695	0.694	0.688	0.690	0.700

original ones.

When comparing BALANCED and UNBALANCED to the third baseline FULL, BALANCED always performs better or same as FULL. regardless of the normalization strategy. It precisely performs exactly the same as FULL in the case of 7300 workers where both approaches result in a full partitioning. On the other hand, UNBALANCED always performs better than FULL in the cases of 50 workers and 500 workers and worse in the case of 7300 workers, using either normalization techniques. Again, this can be mainly attributed to the non-optimality of local decisions made by the greedy UNBALANCED algorithm which might sometimes result in early stopping.

In terms of efficiency, all algorithms finish within seconds (a minimum of

0.001 seconds and a maximum of 26.322 seconds). The algorithm that runs in the least amount of time is obviously FULL since it partitions all workers using all attributes at once without performing any extra checks. On the other hand, the algorithm with the highest running time is UNBALANCED for all cases except for the 7300 workers where BALANCED requires more time regardless of the normalization strategy. This is very intuitive given that UNBALANCED is a recursive algorithm that splits every partition along the way until the stopping condition is met. In the case of 7300 workers however, as we discussed earlier, BALANCED ends up splitting workers on all attributes which results in a deeper partition tree compared to UNBALANCED which stops earlier for most branches.

5.3 Addressing discrimination

Our first goal in this set of experiments is to verify that our normalization-based strategy for addressing discrimination is indeed effective. First, we re-run our maximum discrimination partitioning algorithms BALANCED and UNBALANCED using the normalized functions returned by each algorithm when run on the MovieLens dataset for all three functions f_1 , f_2 and f_3 defined in Section 5.1. Recall that f_1 targets workers with diverse ratings, f_2 targets those with low activity, whereas f_3 does not exhibit discrimination by construct. Table 5.5 displays the KL-divergence of the partitioning with maximum discrimination when we run our two algorithms BALANCED and UNBALANCED using the normalized

scores returned by each algorithm on the original function scores. As can be seen from the table, we find that all normalized functions exhibit less discrimination as indicated by KL-divergence regardless of what algorithm was used, compared to the original scores (see Table 5.1). For instance, when running BALANCED on the normalized data obtained by running BALANCED with f_1 , workers are split on gender only and the KL-divergence is reduced from 0.074 to 0.021. Similarly, when running UNBALANCED on the normalized data obtained by running UNBALANCED on f_2 , the KL-divergence is reduced from 0.080 to 0.028. This is consistent for all other cases. This highlights that by normalizing the function after identifying the partitioning with maximum discrimination, we are able to reduce the amount of discrimination in the data.

Table 5.4: KL-divergence of the partitioning with maximum discrimination for the MovieLens dataset after score normalization using standardization.

Algorithm	BALANCED			UNBALANCED		
	f_1	f_2	f_3	f_1	f_2	f_3
UNBALANCED	0.021	0.028	0.303	0.032	0.015	0.171
BALANCED	0.021	0.028	0.195	0.022	0.024	0.081

Table 5.5: KL-divergence of the partitioning with maximum discrimination for the MovieLens dataset after score normalization using rescaling.

Algorithm	BALANCED			UNBALANCED		
	f_1	f_2	f_3	f_1	f_2	f_3
UNBALANCED	0.047	0.034	0	0.021	0.011	0
BALANCED	0.069	0.034	0	0.029	0.032	0

Our second goal is to identify the impact of addressing discrimination on how workers are targeted by the task assignment function. First, we split our

MovieLens dataset into two sets D_1 and D_2 , where D_1 contains 80% of the ratings for each worker and D_2 contains the remaining 20%. Our goal is to use the data in D_1 to find workers to target and the data in D_2 to verify the usefulness of discrimination quantification and normalization in worker targeting. To this end, we run both BALANCED and UNBALANCED on D_1 , and retrieve the top-100 workers based on the normalized functions f_1 and f_2 . We also retrieve the top-100 workers based on the original functions and the top-100 workers based on a diversity-based strategy that uses the principle of Maximal-Marginal Relevance (MMR) [7] to find the top-100 highest scored workers who are most diverse from each other. The MMR approach re-ranks workers based on their MMR values, which are computed as follows:

$$MMR(w) = \lambda f(w) + (1 - \lambda) \min_{w' \in S} Euclidean(w, w')$$

where $f(w)$ is the function score of worker w , S is the set of workers already selected, $Euclidean(w, w')$ is the Euclidean distance between two workers w and w' , which is computed based on their attributes gender, age, location and occupation, and λ is a weighting parameter, which we set in our experiments to 0.5.

In Table 5.6, we display the average and the standard deviation of the pairwise Euclidean distance between the top-100 workers in each list. As can be seen from the table, *on average the top-100 workers retrieved from the lists that*

Table 5.6: Mean and Standard deviation of the Euclidean distance of the top-100 workers.

List	f_1		f_2	
	Mean	St. Dev.	Mean	St. Dev.
Original	10.416	5.484	11.576	6.046
BALANCED	11.008	5.780	14.032	7.836
UNBALANCED	15.291	8.150	14.589	7.603
BALANCED Rescale	15.087	8.247	13.027	6.721
UNBALANCED Rescale	15.291	8.150	11.576	6.046
MMR	16.323	8.758	16.264	9.197

were generated using our algorithms are more diverse as measured by the pairwise Euclidean distance between the workers than the top-100 workers from the original list and almost as diverse as the top-100 workers retrieved using the MMR approach for both functions. Recall that MMR explicitly uses the Euclidean distance to re-rank workers, and thus it is not surprising that it exhibits higher average pairwise Euclidean distance compared to our algorithms. However, unlike MMR, our approach is fully data-driven, does not involve any parameters, and does not utilize any distance function between workers. On the other hand, the MMR approach requires defining a distance function between workers, which would mean we have to decide which attributes to diversify on before hand. It also involves a weighting parameter to combine distance between workers and their function scores to compute the MMR values. Finally, our approach returns a full ranking of all workers in very short time, whereas MMR requires the value of K , which is the number of top workers to be retrieved after re-ranking, since it will not be feasible to re-rank the set of all workers based on their MMR values.

In our last experiment, we show that while we are capable of diversifying users that were targeted using our algorithms, this does not entail that we compromise the quality of the data acquired. This experiment is very important and shows that targeting workers using our approach also entails higher quality of the data acquired. Table 5.7 displays the average and standard deviation of the variance of the ratings acquired in D_2 by workers targeted in D_1 and their number of ratings in D_2 , which correspond to the functions f_1 and f_2 , respectively. As can be observed from the table, *the top-100 workers targeted by our algorithms have higher rating variance on average compared to the top-100 workers from the original list and the top-100 workers retrieved by the MMR approach*. Moreover, our algorithms when using rescaling would target users whose variances of ratings are comparable to those of the top-100 users from the original list or using the MMR approach. Similarly, our algorithms result in targeting less active workers

Table 5.7: Mean and Standard deviation of the rating variance (f_1) and the number of ratings (f_2) in D_2 for the top-100 workers.

List	f_1		f_2	
	Mean	St. Dev.	Mean	St. Dev.
Original	0.968	0.510	3.990	0.100
BALANCED	0.984	0.506	9.690	9.928
UNBALANCED	0.986	0.548	11.290	11.205
BALANCED Rescale	0.936	0.586	9.330	9.088
UNBALANCED Rescale	0.898	0.604	10.370	10.168
MMR	0.966	0.505	20.180	20.624

in D_2 compared to the MMR approach. Note that when using f_2 as a task assignment function, the top-100 workers in the original list have the fewest

ratings in D_2 because of the way the dataset was split, where 80% of the ratings are in D_1 and 20% in D_2 . This means that the least active workers in both sets D_1 and D_2 would be the same, which explains why the top-100 workers in the original list have the lowest f_2 in D_2 (average of 3.990 and standard deviation of 0.100).

Chapter 6

Conclusion

We tackled the question of unveiling discrimination in crowdsourcing task assignment. We proposed to solve a combinatorial problem that finds a partitioning of workers that exhibits the highest discrimination with respect to a data acquisition process. We developed two heuristics to solve our problem. We showed, on real and simulated datasets, that our heuristics are fast without compromising discrimination values and that score normalization is necessary to acquire less-biased datasets.

The most promising research direction, in our opinion, is to design an interactive human-in-the-loop approach, that unveils discrimination incrementally and involves the worker by suggesting different ways of addressing it. We believe this would achieve a good balance between what the worker wants and unveiling the risks of algorithmic decision-making.

Bibliography

- [1] T. Calders, “Fairness-aware data mining,” in *EGC*, pp. 3–4, 2016.
- [2] J. Stoyanovich, S. Abiteboul, and G. Miklau, “Data responsibly: Fairness, neutrality and transparency in data analysis,” in *EDBT*, pp. 718–719, 2016.
- [3] A. Olteanu, E. Kiciman, and C. Castillo, “A critical review of online social data: Biases, methodological pitfalls, and ethical boundaries,” in *WSDM*, pp. 785–786, 2018.
- [4] M. Noon, “The shackled runner: time to rethink positive discrimination?,” *Work, Employment and Society*, vol. 24, no. 4, pp. 728–739, 2010.
- [5] J. Guynn, “Google photos labeled black people ‘gorillas?’” <https://www.usatoday.com/story/tech/2015/07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465/>, 2015. Online; accessed April 22, 2018.
- [6] J. M. Joyce, “Kullback-leibler divergence,” in *International Encyclopedia of Statistical Science*, pp. 720–722, 2011.
- [7] J. Carbonell and J. Goldstein, “The use of mmr, diversity-based reranking for reordering documents and producing summaries,” in *SIGIR*, pp. 335–336, 1998.
- [8] K. Kirkpatrick, “Battling algorithmic bias: how do we ensure algorithms treat us fairly?,” *Commun. ACM*, vol. 59, pp. 16–17, 2016.
- [9] I. Zliobaite, “A survey on measuring indirect discrimination in machine learning,” *CoRR*, vol. abs/1511.00148, 2015.
- [10] L. Sweeney, “Discrimination in online ad delivery,” *CoRR*, vol. abs/1301.6822, 2013.
- [11] T. Calders and S. Verwer, “Three naive bayes approaches for discrimination-free classification,” *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 277–292, 2010.

- [12] F. Tramèr, V. Atlidakis, R. Geambasu, D. J. Hsu, J. Hubaux, M. Humbert, A. Juels, and H. Lin, “Discovering unwarranted associations in data-driven applications with the fairest testing toolkit,” *CoRR*, vol. abs/1510.02377, 2015.
- [13] J. Pilourdault, S. Amer-Yahia, D. Lee, and S. B. Roy, “Motivation-aware task assignment in crowdsourcing,” in *EDBT*, pp. 246–257, 2017.
- [14] M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi, “Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment,” *arXiv*, 2017.
- [15] C. ju Ho, S. Jabbari, and J. W. Vaughan, “Adaptive task assignment for crowdsourced classification,” in *ICML*, vol. 28, pp. 534–542, 2013.
- [16] D. R. Karger, S. Oh, and D. Shah, “Budget-optimal task allocation for reliable crowdsourcing systems,” *Oper. Res.*, vol. 62, no. 1, pp. 1–24, 2014.
- [17] C.-J. Ho and J. W. Vaughan, “Online task assignment in crowdsourcing markets,” in *AAAI*, pp. 45–51, 2012.
- [18] N. Kaufmann, T. Schulze, and D. Veit, “More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk,” in *AMCIS*, vol. 11, pp. 1–11, 2011.
- [19] B. B. Bederson and A. J. Quinn, “Web workers unite! addressing challenges of online laborers,” in *CHI EA*, pp. 97–106, 2011.
- [20] R. M. Borromeo, T. Laurent, M. Toyama, and S. Amer-Yahia, “Fairness and transparency in crowdsourcing,” in *EDBT*, pp. 466–469, 2017.
- [21] D. Durward, I. Blohm, and J. M. Leimeister, “Is there papa in crowd work?: A literature review on ethical dimensions in crowdsourcing,” in *UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld*, pp. 823–832, 2016.
- [22] S. A. Friedler, C. Scheidegger, and S. Venkatasubramanian, “On the (im)possibility of fairness,” *CoRR*, vol. abs/1609.07236, 2016.
- [23] T. F. Gonzalez, “Clustering to minimize the maximum intercluster distance,” *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.
- [24] D. Jannach, P. Resnick, A. Tuzhilin, and M. Zanker, “Recommender systems - : beyond matrix completion,” *Commun. ACM*, vol. 59, no. 11, pp. 94–102, 2016.
- [25] H. Rahman, S. Thirumuruganathan, S. B. Roy, S. Amer-Yahia, and G. Das, “Worker skill estimation in team-based tasks,” *PVLDB*, vol. 8, no. 11, pp. 1142–1153, 2015.

- [26] J. a. Gama, R. Fernandes, and R. Rocha, “Decision trees for mining data streams,” *Intell. Data Anal.*, vol. 10, no. 1, pp. 23–45, 2006.
- [27] N. Stewart, C. Ungemach, A. J. Harris, D. M. Bartels, B. R. Newell, G. Paolacci, and J. Chandler, “The average laboratory samples a population of 7,300 amazon mechanical turk workers,” *Judgment and Decision making*, vol. 10, no. 5, p. 479, 2015.