Quantifying and Addressing Ranking Disparity in Human-Powered Data Acquisition
Sihem Amer-yahia, Shady Elbassuoni, Ahmad Ghizzawi, Anas Hosami

To cite this version:
Sihem Amer-yahia, Shady Elbassuoni, Ahmad Ghizzawi, Anas Hosami. Quantifying and Addressing Ranking Disparity in Human-Powered Data Acquisition. KDD ’21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Aug 2021, Virtual Event Singapore, France. pp.2525-2533, 10.1145/3447548.3467063 . hal-03379592

HAL Id: hal-03379592
https://hal.archives-ouvertes.fr/hal-03379592
Submitted on 15 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Quantifying and Addressing Ranking Disparity in Human-Powered Data Acquisition

Authors’ Copy

Sihem Amer-Yahia
CNRS, University of Grenoble Alpes, France
sihem.amer-yahia@univ-grenoble-alpes.fr

Ahmad Ghizzawi
American University of Beirut, Lebanon
ahg05@mail.aub.edu

Shady Elbassuoni
American University of Beirut, Lebanon
se58@aub.edu.lb

Anas Hosami
American University of Beirut, Lebanon
amh89@mail.aub.edu

ABSTRACT
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 human users to generate training data. The generated data might be biased if the data acquisition process is skewed towards certain groups of people based on say gender, ethnicity or location. This typically happens as a result of a hidden association between the people’s qualifications for data acquisition and the people’s protected attributes. In this paper, we study how to unveil and address disparity in data acquisition. We focus on the case where the data acquisition process involves ranking of people and we define disparity as the unbalanced targeting of people by the data acquisition process. To quantify disparity, we formulate an optimization problem that partitions people on their protected attributes, computes the qualifications of people in each partition, and finds the partitioning that exhibits the highest disparity in qualifications. Due to the combinatorial nature of our problem, we devise heuristics to navigate the space of partitions. We also discuss how to address disparity between partitions. We conduct a series of experiments on real and simulated datasets that demonstrate that our proposed approach is successful in quantifying and addressing ranking disparity in human-powered data acquisition.

CCS CONCEPTS
• Information systems → Data management systems; • Human-centered computing → Collaborative and social computing.

KEYWORDS
fairness, disparity, ranking, data acquisition

1 INTRODUCTION
A major source of algorithmic bias in AI is training data [6, 30, 34] and many AI applications rely on humans to generate that data. If the data acquisition process is skewed towards certain groups of people based on say gender, ethnicity or location, the model trained with that data is likely to be biased. Even if the data acquisition process explicitly targets some groups, e.g., in the case of positive discrimination [29], data may be biased with respect to subgroups within these groups. This typically happens as a result of a hidden association between people’s qualifications for data acquisition and people’s protected attributes. This association can be intentional or unintentional. In both cases, it favors acquiring data from certain groups of people over others. The ability to detect such associations is a necessary first step towards ensuring fairness in decision-making. In this paper, we are interested in unveiling, quantifying, and addressing disparity in human-powered data acquisition.

Human-powered data acquisition fuels many applications on the Web. It is for instance at the heart of the social Web where millions of people volunteer their opinions in the form of posts, tags, ratings and reviews. It is also central to crowdsourcing platforms that are frequently used for cheap and fast data acquisition (Appen is specifically used for labeling data for instance). On collaborative rating sites such as MovieLens, people discover items to rate through a recommendation mechanism that decides which people are most likely to rate which items. By rating items, they contribute implicitly to data acquisition. In crowdsourcing, data is requested explicitly via Human Intelligence Tasks (HITs) and to ensure high-quality data, requesters rely on qualification tests. In this work, we are interested in studying disparity in implicit and explicit data acquisition. We abstract that process with a scoring function used to select people for a task; rating an item, or completing a HIT. In collaborative rating, the scoring function could target people located in some region. In crowdsourcing, a typical function

1https://appen.com/
2https://movielens.org/
is one that combines one’s acceptance ratio on the platform and performance on sample HITs.

While scoring functions can filter out unqualified people, they might also result in a skewed distribution of people that are targeted by a task. For instance, it might be the case that while not intentional, the majority of people that are allowed to attempt a task are males, white, young, or combinations of those. This is due to a hidden association between the scoring function and the people’s protected attributes such as gender, ethnicity and age. Without identifying such associations, one runs the risk of acquiring biased data and training biased models (e.g., the Google image classifier was showing systematic bias in recognizing images of African Americans [20]). To date, there is no framework to characterize people who are targeted by data acquisition and to help reduce the risk of acquiring biased data.

**Quantifying Disparity.** We define disparity as the unbalanced targeting of people by the data acquisition process and advocate the need for an algorithmic approach for unveiling and quantifying it. To do so, we propose to discover groups or partitions of people targeted by a data acquisition process using any combination of their protected attributes, in a data-driven fashion rather than assume the groups of interest are given as in most related work. More precisely, for each possible partitioning of people using their protected attributes, we must examine the scores assigned to people by the data acquisition scoring function, and quantify the difference in scores across partitions. Since the number of ways to partition people is combinatorial in the number of protected attributes, we propose to model an optimization problem that finds a partitioning of people, for which the data acquisition process exhibits the highest disparity. The rationale is that the partitioning with the highest disparity will subsume all others.

**Addressing Disparity.** One possible way to address disparity is to normalize the obtained scores across the identified partitions, i.e., those that exhibit the highest disparity, to make partitions comparable. This is borrowed from the machine learning community where normalization is commonly used to make features comparable without distorting differences in the ranges of values. We refer to this approach as normalization-based. Consequently, we can then choose the K highest scoring people after normalization or apply a threshold to filter out less-qualified people for instance. Of course, other normalizations could be proposed and tested. Our approach for addressing disparity thus aims to mitigate disparate treatment across multiple demographic groups by taking into account the score of the individuals (i.e., utility) in those different demographic groups while exposing people from different groups (i.e., exposure). In other words, our approach aims to diversify the people that participate in the data acquisition process based on their protected attributes and qualifications for the data acquisition process.

**Empirical Evaluation.** We conduct experiments on real and simulated datasets to show the effectiveness of our approach in quantifying and addressing disparity in human-powered data acquisition. We first show that our approach is successful in identifying the maximum disparity of several data acquisition processes on a MovieLens dataset. We also show that by addressing the identified disparity using our normalization-based strategy, the resulting normalized scoring function will exhibit less disparity. We also demonstrate that our normalization-based strategy is comparable to a baseline diversity-based strategy as it achieves high representativeness when acquiring ratings from people in MovieLens, without unduly sacrificing the quality of the acquired ratings. Finally, we validate that our heuristics-based algorithm is more successful in identifying maximum disparity on simulated data compared to baseline algorithms and to an optimum algorithm that solves our optimization problem exactly.

To summarize, we make the following contributions:

1. We define disparity as the unbalanced targeting of people by the data acquisition process based on their protected attributes. We argue that quantifying and addressing disparity is necessary even in the case where some person’s protected attributes are used in the acquisition process (i.e., in the case of positive discrimination).
2. We formalize disparity quantification in a data-driven fashion as the largest difference between an original ranking of people and a per-partition normalized ranking. Due to the combinatorial number of partitionings, we devise heuristics to compute disparity in acceptable time and propose a normalization-based strategy to address it.
3. We conduct experiments on real and simulated datasets. Our results show that our approach is effective in quantifying and addressing disparity of various data acquisition processes without negatively affecting their quality. They also show that our algorithm strikes a good balance between effectiveness and efficiency when quantifying and addressing disparity compared to its baselines and an optimum algorithm.

### 2 MOTIVATING EXAMPLES

**Tweet Sentiment Annotation.** Consider a data acquisition scenario that gathers training data for tweet sentiment annotation. Each task consists of annotating a single tweet and some guidelines to help people achieve the task. A typical scoring function is the combination of acceptance ratio and a qualification test, where the qualification test assesses the performance of a person on sample tweets with ground truth. Naturally, only qualifying people with a score above a certain threshold will be selected to attempt the task. Assume that people have only two binary protected attributes: age = {young, old} and country = {USA, India}. Also assume that the scoring function is only above the threshold for young people from India and older people from the USA, and thus only those people are allowed to attempt the tasks. The scoring function will thus appear fair when one considers either protected attribute alone, in the sense that it allows both young and old people to attempt the tasks, and also allows both Indians and Americans to attempt the tasks. However, if one looks at the conjunction of these two attributes, it is clear that the scoring function is unfair with respect to old Indians or young Americans. This can be attributed for instance to a correlation between age and country and the acceptance ratio or qualification test scores. Our goal in this paper is to unveil disparity between person groups in cases like this. This requires examining all possible combinations of protected attributes to quantify such disparity, i.e., at which proportions does each person group get assigned the annotation task and how different groups are targeted.
(e.g., old people in India will be less likely allowed to attempt the tasks than younger people from India).

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

**Ratings of American Blockbusters.** Consider a scenario that explicitly targets Europeans to gather diverse ratings on American blockbusters. To achieve that, the scoring function is applied to Europeans and computes their rating variance to target people whose ratings are diverse. Consider two protected attributes: Gender = {male, female} and country = {Spain, France, USA}. Since the task targets only Europeans, the case of excluding Americans is an example of positive discrimination. However, disparity can still occur within subgroups of Europeans: e.g., women whose rating variance is generally lower than men, or, French people whose ratings are harsher than Spanish people. The ability to unveil that disparity will shed light on the hidden associations between rating variance and subgroups, 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.

## 3 DATA MODEL

We present our data model and define the problem of unveiling and quantifying disparity. We are given a set of protected attributes $A = \{a_1, a_2, \ldots, a_n\}$ and a set of qualifications $B = \{b_1, b_2, \ldots, b_m\}$. Attributes in $A$ are inherent properties of people such as gender, age, ethnicity, origin, etc. Qualifications in $B$ represent the abilities of a person for data acquisition tasks. In crowdsourcing, qualifications include the acceptance ratio of the person, 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 person for a movie or the opinion of a person about a restaurant. Qualifications may be explicitly given by people or inferred from previously rated items as in recommendation strategies [23], or from previously completed tasks in crowdsourcing [31].

A data acquisition process $D = (\text{task}, f)$ contains a task, such as annotating a tweet or rating a movie, and a scoring function $f : U \rightarrow R$ that calculates for a person $u \in U$ a qualification score for the task. For instance, for tweet annotation, $f$ could simply be the location and language skill of the person or a more sophisticated formula that aggregates the person’s acceptability on the platform, the quality of the person’s past contributions, and the person’s language skill. For a movie rating task, $f$ could be the variance of ratings of the person, or a sophisticated procedure such as a recommendation strategy that computes the expected rating of the person for a movie.

The scoring function $f$ can make use of any attributes in $A$ and qualifications in $B$. Its exact shape is not important for the purpose of this work. Our goal is to quantify the disparity that happens as a result of applying $f$ to people in $U$ for a given data acquisition task in $D$. People in $U$ can be sorted in increasing or decreasing order of their scores computed by $f$. We refer to the resulting list as $L_0$. To quantify disparity induced by $f$, we consider a full disjoint partitioning $P = \{p_1, p_2, \ldots, p_k\}$ of the set of people $U$ on their attributes in $A$. Each person must belong to one and only one partition $p_i$ and each partition $p_i$ is obtained using the same set of attributes. Given a person $u \in p_i$, we define $f'(u)$ as the normalized score of person $u$ in partition $p_i$. We experiment with two methods of normalization, namely standardization and rescaling. In the former, $f'(u)$ is computed as follows:

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

where

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

and

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

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

$$f'(u) = \frac{f(u) - \min_{u \in p_i} f(u)}{\max_{u \in p_i} f(u) - \min_{u \in p_i} f(u)}$$

where

$$\min_{u \in p_i} f(u)$$

and

$$\max_{u \in p_i} f(u)$$

In case a partition $p_i$ consists of only one person $u$, the normalized score of that person $f'(u)$ in partition $p_i$ would be set to the original score $f(u)$ normalized with respect to all people considered for the data acquisition task using either of the normalization methods listed above.

We rank people $u \in U$ based on their normalized function scores $f'(u)$ to obtain a new ranking of people $L_P$. Finally, we measure disparity of the partitioning $P$ as the *divergence* of the ranking of people after per-partition normalization $L_P$ from the original ranking of people $L_0$, which we denote as $\Delta(L_P||L_0)$. A variety of functions can be used to compute $\Delta$ including Kendall-Tau, Normalized Discounted Cumulative Gain (NDCG) [24], and Kullback-Leibler divergence (KL-divergence) [25], which is the one we adopt in our experiments. The scoring function $f$ is said to not exhibit disparity on people $U$, if and only if there does not exist any full partitioning $P$ of people $U$ such that $\Delta(L_P||L_0) \neq 0$. Intuitively, the higher the value of $\Delta$, the higher the disparity of the data acquisition process.

**Example.** For example, consider the following data acquisition process in a crowdsourcing setting. Table 1 displays a set of people $U$ consisting of 10 people, 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 people $u \in U$ as follows: $f(u) = 0.3 \times \text{LanguageTest}(u) + 0.7 \times \text{ApprovalRate}(u)$. The first row of Table 2 shows the original ranked list of people $L_0$ based on their $f$ values. The second and third rows show the ranked lists of people obtained from two different partitionings $P_1$ and $P_2$, and the quantified disparity of each partitioning using KL-divergence as $\Delta$ between $L_P$ and $L_0$, $L_P$ and $L_0$ respectively. The partitionings $P_1$ and $P_2$ are displayed in Figures 1 and 2. For $P_1$, the people are partitioned based on language first then gender. For $P_2$, the people are partitioned on country first, and then year of birth. The leaf nodes represent the final partitions in $P_1$ and $P_2$. 

---

1 We will drop the word protected henceforth unless it is not clear from the context.
Our disparity quantification problem is hence the problem of finding a full disjoint partitioning \( P \) of people in \( U \). One approach is to consider all possible partitionings of people based on their attributes and retrieve the partitioning that returns the maximum disparity as measured by \( \Delta \) between the final ranking of people \( L_P \) induced by per-partition normalization and the original ranking of people \( L_O \). Intuitively, the partitioning with the maximum disparity is the one that captures the most bias induced by the data acquisition process \( D \). Finding such a partitioning constitutes our optimization problem that we formulate as follows.

**Definition 1 (Max Disparity Partitioning).** Given a set of people \( U \) and a data acquisition process \( D = (\text{task}, f) \), our goal is to fully partition people in \( U \) into disjoint partitions \( P = \{p_1, p_2, \ldots, p_k\} \) based on their attributes in \( A \) using the following optimization objective:

\[
\arg\max_P \Delta(L_P || L_O)
\]

subject to \( \forall i,j \ p_i \cap p_j = \emptyset \)
\[
\bigcup_{i=1}^{k} p_i = U
\]

where \( \Delta(L_P || L_O) \) is a disparity measure between the two ranked lists \( L_P \) and \( L_O \).

For instance, using KL-divergence as the disparity measure in the above formulation, we have:

\[
\Delta(L_P || L_O) = \sum_{u \in U} r_p(u) \log \frac{r_p(u)}{r_O(u)}
\]

where \( r_O(u) \) is the rank of person \( u \) in \( L_O \) and \( r_P(u) \) is the rank of \( u \) in \( L_P \).

It is obvious that our problem for finding the maximum disparity partitioning is hard since there are many possible partitionings. For instance, assuming that the number of attributes is \( n \), then the number of possible ways to partition the people is \( O(2^n) \), which can be extremely large even for small values of \( n \). For this reason, in the next section, we propose to develop a heuristics-based algorithm to identify partitionings of people with respect to our optimization objective within reasonable time. We will also describe how we propose to address disparity once it is quantified by our algorithm.

### 4 APPROACH

#### 4.1 Quantifying Disparity

As explained in the previous section, to quantify disparity we rely on solving an optimization problem that finds a partitioning of the people that maximizes disparity. Our optimization problem is hard due to the exponential number of possible partitionings. For this reason, we propose to use a heuristics-based algorithm to identify the partitioning of people with the highest disparity. Our algorithm is a greedy algorithm that relies on local decisions to maximize disparity. It relies on the same principle as decision partition trees that use a gain function to split a dataset [17]. In our case, the gain function relies on measuring disparity between rankings.

Algorithm 1 shows the pseudocode for our algorithm MDP, which stands for **Maximum Disparity Partitioner**. It takes as input a set of people \( U \), a scoring function \( f : U \rightarrow R \) and a set of protected attributes \( A \). It returns a partitioning \( P \) of all people in \( U \). MDP starts by one partition containing all people in \( U \). It attempts to split that partition on the attribute that results in the highest disparity between the normalized ranking of people after the split and the current ranking as measured by \( \Delta \). It then repeatedly tries to split people on the remaining attributes and only stops when the disparity 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 partitions in current as the obtained partitioning \( P \), which is used to generate its final ranked list \( L_P \).

Algorithm MDP makes use of two helper methods normalize() and highestAttribute(). normalize() takes a set of partitions (i.e., a partitioning) and a scoring function, normalizes each partition using either standardization (mean and standard deviation) or rescaling.
where

with the highest disparity (i.e., \((\text{MIN-MAX})\) and returns a ranked list of the people after their scores are normalized. \(\text{highestAttribute()}\) takes a set of partitions, a scoring function and a set of attributes. It returns the attribute with the highest disparity (i.e., \(A\) value) between the current ranking of people based on the given partitions and the new ranking of people after they are split using that attribute and performing a per-partition normalization of scores.

Algorithm MDP has a complexity of \(O(n^2)\) in the worst case, where \(n\) is the number of protected attributes. At first the algorithm tries out \(n\) possible partitionings using a single attribute, and then it tries \(n - 1\) partitionings corresponding to the remaining \(n - 1\) attributes and so on until there are no more attributes left. Hence, it will examine a total of \(n + (n - 1) + \cdots + 1 = O(n^2)\) partitionings in the worst case (i.e., if the termination condition was never met).

4.2 Addressing Disparity

Once a partitioning \(P\) is obtained, we propose to address disparity as follows. First, we normalize the people’s function scores for each partition \(p \in P\) using one of the normalization techniques described in Section 3 to obtain new function scores \(f'(u)\). We then re-rank all people in all partitions globally based on their new scores \(f'(u)\) and return the new ranked list \(LP_2\) along with the new scores of people \(f'(u)\) obtained after normalization.

For example, consider our toy example from the previous section shown in Table 1. Assume that our algorithm returned \(P_2\) as the partition with the highest disparity, which is shown in Figure 2. To address disparity, we first normalize the scores in each partition in \(P_2\) and then we re-rank all the people in all partitions based on their normalized scores. Finally, we return the new ranked list \(LP_2\) as the new ranked list for the data acquisition process. The new ranked list \(LP_2\) is shown as the third row in Table 2.

5 EXPERIMENTS

We conduct experiments on real and simulated datasets. Our first set of experiments constitutes a proof of concept for the need to study and address disparity of different acquisition processes on a MovieLens dataset. Our second set of experiments is used to validate our heuristics in identifying maximum disparity using simulated data. In all experiments, we use the Kullback-Leibler divergence (KL-divergence) [25] to implement the disparity measure \(\Delta\) introduced in Section 3.

5.1 MovieLens Dataset

We first evaluate our approach on the MovieLens 1M dataset\(^4\). The dataset consists of 6,040 people with around 1 million ratings in total (972,599 to be exact). Each person 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 data acquisition processes. The first process aims to acquire ratings from people who have a diverse set of ratings. To this end, we use rating variance as the scoring

\(^4\)https://grouplens.org/datasets/movielens/1m/
Table 3: KL-divergence of the partitioning with maximum disparity for the MovieLens dataset.

<table>
<thead>
<tr>
<th>Normalization Technique</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardization</td>
<td>0.074</td>
<td>0.077</td>
<td>0.470</td>
</tr>
<tr>
<td>Rescaling</td>
<td>0.103</td>
<td>0.096</td>
<td>0.000</td>
</tr>
</tbody>
</table>

function, i.e.,:

$$f_1(u) = \frac{1}{|I_u|} \sum_{i \in I_u} (r_u(i) - \mu_u)^2$$

where $I_u$ is the set of movies that person $u$ rated, $r_u(i)$ is the rating provided by $u$ for movie $i$, and $\mu_u$ is the average of all ratings provided by $u$. To discard people with a very small number of ratings, we only consider people who have rated more than 100 movies. The second process aims to acquire ratings from the least active people. We use a scoring function $f_2$ to rank people based on the number of movies they rated. Our third process does not exhibit any disparity by construct. It relies on a scoring function $f_3$ that takes a person and returns 0 if the person has rated fewer than 20 movies and 1 otherwise. The reason we opted for a threshold of 20 is that MovieLens requires at least 20 ratings per person on their website. Nonetheless, some of the people in the dataset have fewer than 20 rating.

Quantifying Disparity. In this first experiment, our goal is to answer the following question: given a dataset of people and a data acquisition process, how much disparity does this process exhibit when treating different groups of people? Given one of the three scoring functions $f_1$, $f_2$, and $f_3$, and report the obtained KL-divergence in Table 3. In the case of rescaling (third row of the table), our algorithm unveiled some disparity for the first two functions $f_1$ and $f_2$ and no disparity 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 (second row of the table), our algorithm unveiled some disparity 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 people). That is, if two partitions have different sizes but the same score distribution, the scores after normalization end up being different. In general, when people are not evenly distributed with respect to their attributes, e.g., there are many more younger people than older ones, using rescaling normalization might be more effective in quantifying disparity as it is less sensitive to the sizes of the partitions. Thus, in the rest of this section, we just present the results for normalization using rescaling.

Addressing Disparity. In this subset of experiments, our goal is to answer the following question: does addressing disparity using our normalization based strategy reduce the disparity of a data acquisition process? To answer this question, we conduct three experiments. The first experiment shows that score normalization reduces data disparity. The second experiment shows that our approach leads to a more diversified set of people, compared to the original ranking. Finally, in our third experiment, we show that targeting people using our approach does not unduly affect the quality of the data acquired.

In our first experiment, we re-run MDP using the normalized values returned by the same algorithm for each of the three functions $f_1$, $f_2$, and $f_3$. Recall that $f_1$ targets people with diverse ratings, $f_2$ targets those with low activity, whereas $f_3$ does not exhibit disparity by construct. Table 4 displays the KL-divergence of the partitioning with maximum disparity when we re-run MDP using the normalized scores returned by the algorithm on the original function scores. As can be seen from the table, we find that all normalized functions exhibit less disparity as indicated by KL-divergence, compared to the original scores. For instance, when running MDP on the normalized data obtained by running MDP with $f_2$, people are split on age only and the KL-divergence is reduced from 0.096 to 0.069. This is consistent for all other functions and highlights that by normalizing scores after identifying the partitioning with maximum disparity, we are able to reduce the amount of disparity of the data acquisition process. Hence, normalization is a good way to address disparity.

In the second experiment, we split our MovieLens dataset into two sets $D_1$ and $D_2$, where $D_1$ contains 80% of the ratings for each person and $D_2$ contains the remaining 20%. Our goal is to use the data in $D_1$ to find people to target and the data in $D_2$ to verify the usefulness of disparity quantification and normalization in targeting people. To this end, we ran MDP on $D_1$ and retrieved the top-100 people based on the normalized functions $f_1$ and $f_2$. We also retrieved the top-100 people based on the original functions. Finally, we used a diversity-based strategy that relies on the principle of Maximal-Marginal Relevance (MMR) [8] to find the top-100 highest scored people who are most diverse from each other. The MMR approach re-ranks people based on their MMR values, which are computed as follows:

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

where $f(u)$ is the function score of person $u$, $S$ is the set of people already selected, $\text{Euclidean}(u, u')$ is the Euclidean distance between two people $u$ and $u'$, which is computed based on their attributes gender, age, location and occupation, and $\lambda$ is a weighting parameter, which we set in our experiments to 0.5.

In Table 5, we display the mean and standard deviation of the pairwise Euclidean distance between the top-100 people based on their demographics in each list over five different runs of the experiment. As can be seen from the table, on average the top-100 people retrieved from the lists that were generated using our algorithm are more diverse as measured by the pairwise Euclidean distance

Table 4: KL-divergence of the partitioning with maximum disparity for the MovieLens dataset after score normalization.

<table>
<thead>
<tr>
<th>scores</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>0.103</td>
<td>0.096</td>
<td>0.000</td>
</tr>
<tr>
<td>Normalized</td>
<td>0.047</td>
<td>0.069</td>
<td>0.000</td>
</tr>
</tbody>
</table>
5.2 Simulated Datasets

The goal of this set of experiments is to evaluate our heuristics-based algorithm, MDP, for quantifying maximum disparity, and to compare it to an optimum algorithm. To do this, we simulate a crowdsourcing platform consisting of 100,000 people. We then sample three different datasets of active people from the platform. The first dataset consists of 50 active people (i.e., \(|U| = 50\)). The second consists of 500 active people and the third consists of 7,300 active people, which is estimated to be the size of active people on Amazon Mechanical Turk [33]. Each \(u \in U\) 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 person are set at random. We also generate five scoring functions \(f_i(u)\) that score people based on their qualifications as follows:

\[ f_i(u) = a_i q_1 + (1 - a_i)q_2 \]

where \(1 \leq i \leq 5\) and \(a_i \in [0,0.3,0.5,0.7,1]\). That is, each scoring function is defined as a linear combination of the two qualifications. Of course, other types of functions could also be used, but our goal in this simulation as mentioned above is to compare our approach to baselines and an optimum algorithm, and thus the shape of the scoring function itself would not make a difference in the comparison.

We compare MDP to two baselines. The first is a copy of our algorithm, which we refer to as \(r\)-MDP and which uses 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. \(r\)-MDP is used to validate our splitting heuristic that greedily picks the attribute that results in the highest KL-divergence. The second baseline is an algorithm that partitions people 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. Table 7 displays the KL-divergence between the original ranking of people and the ranking after normalizing their scores in each of the identified partitions using rescaling, and the average runtime of the algorithms over five runs. In the table, we also show the performance of an optimum algorithm which exhaustively tries out every possible partitioning of people and returns the one with the highest disparity (i.e., one that solves our optimization problem exactly).

As can be seen from Table 7, in the majority of cases, MDP outperforms both baselines by finding a higher KL-divergence. It performs worse than its random counterpart \(r\)-MDP in terms of KL-divergence only 4 times out of 15. 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 people and the original ones. When comparing MDP to the second baseline Full, MDP always performs better or same as Full. It precisely performs exactly the same as Full in the case of 7,300 people where both approaches result in a full partitioning. Finally, when comparing

Table 5: Mean and Standard deviation of the Euclidean distance of the top-100 people.

<table>
<thead>
<tr>
<th>List</th>
<th>(f_1) MEAN</th>
<th>(f_1) St. Dev.</th>
<th>(f_2) MEAN</th>
<th>(f_2) St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>12.340</td>
<td>7.267</td>
<td>11.945</td>
<td>6.513</td>
</tr>
<tr>
<td>MDP</td>
<td>12.478</td>
<td>6.388</td>
<td>13.472</td>
<td>7.349</td>
</tr>
<tr>
<td>MMR</td>
<td>20.254</td>
<td>11.276</td>
<td>21.607</td>
<td>11.903</td>
</tr>
</tbody>
</table>

Table 6: Mean and Standard deviation of the rating variance \((f_1)\) and the number of ratings \((f_2)\) in \(D_2\) for the top-100 people.

<table>
<thead>
<tr>
<th>List</th>
<th>(f_1) MEAN</th>
<th>(f_1) St. Dev.</th>
<th>(f_2) MEAN</th>
<th>(f_2) St. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>1.962</td>
<td>0.395</td>
<td>4.140</td>
<td>0.347</td>
</tr>
<tr>
<td>MDP</td>
<td>1.441</td>
<td>0.477</td>
<td>9.110</td>
<td>7.357</td>
</tr>
<tr>
<td>MMR</td>
<td>1.529</td>
<td>0.573</td>
<td>18.513</td>
<td>23.4399</td>
</tr>
</tbody>
</table>

between the people than the top-100 people from the original list. On the other hand, MMR achieved the highest diversity in terms of people’s attributes for both functions. Recall that MMR explicitly uses the Euclidean distance to re-rank people, and thus it is not surprising that it exhibits the highest average pairwise Euclidean distance. However, unlike MMR, our approach is fully data-driven, does not involve any parameters, and does not utilize any distance function between people. Additionally, MMR requires defining a distance function between people, which would mean we have to decide which attributes to diversify on beforehand. It also involves a weighting parameter to combine distance between people and their function scores to compute the MMR values. Finally, our approach returns a full ranking of all people in a very short time, whereas MMR requires the value of \(K\), which is the number of top people to be retrieved after re-ranking, since it will not be feasible to re-rank the set of all people based on their MMR values.

In our third experiment in this subset, we show that targeting people with our approach does not unduly affect the quality of the data acquired. Table 6 displays the mean and standard deviation of 1) the variance of the ratings acquired in \(D_2\) by people targeted in \(D_1\), and 2) 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, our algorithm results in targeting less active people in \(D_2\) compared to MMR. Note that when using \(f_2\), the top-100 people 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 people in both \(D_1\) and \(D_2\) would be the same, which explains why the top-100 people in the original list have the lowest \(f_2\) in \(D_2\) (average of 4.140 and standard deviation of 0.337). On the other hand, the top-100 people in the original list had slightly higher rating variance on average compared to the top-100 people from the lists generated by our algorithm and the top-100 people retrieved by MMR. This highlights the tradeoff between eliminating disparity and optimizing the data acquisition scoring function.
Table 7: KL-divergence of the partitioning with maximum disparity and the time taken to identify that partitioning.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>KL-divergence</th>
<th>Time (in secs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>MDP</td>
<td>0.086</td>
<td>0.094</td>
</tr>
<tr>
<td>r-MDP</td>
<td>0.168</td>
<td>0.125</td>
</tr>
<tr>
<td>Full</td>
<td>0.019</td>
<td>0.010</td>
</tr>
<tr>
<td>Optimum</td>
<td>0.207</td>
<td>0.215</td>
</tr>
</tbody>
</table>

Results for 50 people

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>KL-divergence</th>
<th>Time (in secs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDP</td>
<td>0.202</td>
<td>0.103</td>
</tr>
<tr>
<td>r-MDP</td>
<td>0.163</td>
<td>0.073</td>
</tr>
<tr>
<td>Full</td>
<td>0.075</td>
<td>0.066</td>
</tr>
<tr>
<td>Optimum</td>
<td>0.202</td>
<td>0.175</td>
</tr>
</tbody>
</table>

Results for 7,300 people

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>KL-divergence</th>
<th>Time (in secs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDP</td>
<td>0.144</td>
<td>0.144</td>
</tr>
<tr>
<td>r-MDP</td>
<td>0.115</td>
<td>0.115</td>
</tr>
<tr>
<td>Full</td>
<td>0.144</td>
<td>0.144</td>
</tr>
<tr>
<td>Optimum</td>
<td>0.144</td>
<td>0.144</td>
</tr>
</tbody>
</table>

MDP to Optimum, they both identify the same optimum disparity 8 times out of 15.

In terms of efficiency, all algorithms finish within seconds (a minimum of 0.005 seconds and a maximum of 29.825 seconds), except for optimum of course. The algorithm that runs in the least amount of time is obviously Full since it partitions all people using all attributes at once without performing any extra checks. On the other hand, the algorithm with the highest running time is Optimum for all cases as it has to examine an exponential number of partitionings to identify the optimum one. Finally, r-MDP is obviously faster than MDP as it uses a random attribute in each iteration to split on rather than examining all remaining attributes to determine which one will result in higher disparity as in the case of MDP.

6 RELATED WORK

Algorithms have replaced and outdone humans in many tasks but they often take biased decisions [7, 16, 27, 35, 44, 44]. To detect discrimination in algorithms, a framework [36] 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 disparity in treatment between worker partitions defined using a combination of multiple protected attributes.

There is a wealth of work on addressing discrimination in machine learning. Most of these works, however, aim to ensure that the output of the machine learned model (typically a classifier) does not discriminate against one or more protected groups (see for instance [15, 21, 22, 26, 39, 41, 42]). Our goal in this paper is different, since we aim to quantify and address disparity at the stage of data acquisition. Our work can thus be viewed as complementary or as an initial step to guarantee the acquisition of less biased training data. There is also a wealth of work on addressing fairness of ranking in general (for example [9, 32, 38, 40]). Unlike our work, the majority of these works that focus on group fairness either assume the presence of predefined groups based on protected attributes of people, or the presence of ranking constraints that bound the number of people per protected attribute value in the top-k ranking. On the other hand, the work in [3] focuses on addressing amortized individual fairness in a series of rankings.

In [28], the authors suggest that to reduce discrimination in crowdsourcing, platforms should track the composition of their worker population, and experiment with their algorithms and datasets. Several discrimination scenarios in crowdsourcing were defined in [5]. In [13], 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 crowdworkers, which addresses their working conditions and benefits. In [10], the authors studied trending topics on Twitter and showed that traditionally marginalized social groups (e.g., black women) are systematically under-represented among the promoters of Twitter trends. This indeed advocates for new algorithms that explicitly take into account the demographic biases of the crowds from which data is acquired, which is what we proposed to do here.

Diversity is a widely studied subject that finds its roots in Web search and databases. Most approaches fall into two cases: content-based (e.g., [8]) and intent-based (e.g., [11]). Gollapudi et al. [18] adopt an axiomatic approach to diversity to address user intent. They show that no diversity function can satisfy all axioms together. Other content-based functions, such as ones based on taxonomies, are possible [2]. In the database context, Chen and Li [12] propose to post-process structured query results to enforce diversity. Similarly, in recommendations [14, 43], intermediate results are post-processed, using pairwise item similarity, to generate accurate
and diverse recommendations. In [37], a hierarchical notion of diversity in databases is introduced, and efficient top-k processing algorithms are proposed. In [1], an algorithm with provable approximation guarantees is proposed to find relevant and diverse news articles.

7 CONCLUSION AND FUTURE WORK

We tackled the question of unveiling ranking disparity in human-powered data acquisition. We proposed to solve a combinatorial problem that finds a partitioning of people that exhibits the highest disparity with respect to a data acquisition process. We developed a heuristics-based decision-tree style algorithm to efficiently solve our optimization problem. We showed, on real and simulated datasets, that our heuristics are fast without compromising disparity values and that score normalization is necessary to acquire less-biased datasets.

Next, we aim to design an interactive human-in-the-loop approach, that unveils disparity incrementally and involves people by suggesting different ways of addressing it. We are currently investigating this approach for several use cases crafted by experts in Business and Management. We are also experimenting with other metrics instead of Kullback-Leibler divergence to measure disparity and solving this approach for several use cases crafted by experts in Business and Management.

ACKNOWLEDGMENTS

This work was funded by the American University of Beirut Research Board (URB), award number 103951.

REFERENCES