MithraCoverage: A System for Investigating Population Bias for Intersectional Fairness

Zhongjun Jin
University of Michigan
markjin@umich.edu

Mengjing Xu
University of Michigan
mengjinx@umich.edu

Chenkai Sun
University of Michigan
sunchenk@umich.edu

Abolfazl Asudeh
University of Illinois at Chicago
asudeh@uic.edu

H. V. Jagadish
University of Michigan
jag@umich.edu

ABSTRACT

Data-driven technologies are only as good as the data they work with. On the other hand, data scientists have often limited control on how the data is collected. Failing to contain adequate number of instances from minority (sub)groups, known as population bias, is a major reason for model unfairness and disparate performance across different groups. We demonstrate MithraCoverage, a system for investigating population bias over the intersection of multiple attributes. We use the concept of *coverage* for identifying intersectional subgroups with inadequate representation in the dataset. MithraCoverage is a web application with an interactive visual interface that allows data scientists to explore the dataset and identify subgroups with poor *coverage*.

CCS CONCEPTS

• **Mathematics of computing** → *Exploratory data analysis*; • **Information systems** → *Data cleaning*; *Data mining*; • **Theory of computation** → *Incomplete, inconsistent, and uncertain databases.*

KEYWORDS

Fairness; Data Ethics; Responsible Data Science

ACM Reference Format:
Zhongjun Jin, Mengjing Xu, Chenkai Sun, Abolfazl Asudeh, and H. V. Jagadish. 2020. MithraCoverage: A System for Investigating Population Bias for Intersectional Fairness. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD'20), June 14–19, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3318464.3384689

1 INTRODUCTION

Big data technologies have affected every aspect of human life and society. These technologies have made our lives unimaginably more shared, connected, convenient, and cost-effective. Nevertheless, *an algorithm is only as good as the data it works with* [4].

An essential piece to training machine learning models is data. However, a data scientist typically has limited, or no, control over how the data has been collected. As a result, he or she conducts analyses on what is available as "found data". It has recently been recognized that, in addition to represent the underlying distribution, the data must include enough examples from less popular "categories", if these categories are to be handled well by the system. Failing to include enough samples from minorities is considered *population bias* [12]. Population bias in the training data can result in *unfair models* that perform differently across different groups. This unfairness may be particularly consequential with social data, where groups may be based on attributes such as race, gender, socio-economic status, education level and so on.

A well-known incident underlining this is the case of the "Google gorilla" [11] where an early image recognition algorithm released by Google had not been trained on enough dark-skinned faces. Although performing near perfectly for light-skinned faces, when presented with an image of a female African American, the algorithm labeled her as a "gorilla". The Google incident is not unique. Similar incidents happened for Nikon [13] and many more. Disparate model behavior becomes even more critical when those are used for data-driven algorithmic decision making. Consider a tool designed to predict how likely an individual is to recidivate. Of course, providing insightful signals to judges, such models can help making societies safer and, hence, are popular in the judicial system. On the other hand, wrong signals can impact individuals' lives at an unprecedented scale [2].

SIGMOD'20, June 14–19, 2020, Portland, OR, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-6735-6/20/06…$15.00
https://doi.org/10.1145/3318464.3384689

Even though Google decided to "ban gorillas" [9], a better solution would be to ensure that the training data has enough entries in each "group". Identifying population bias on single attributes such as race is straight-forward. However, in general the groups can be defined as the *intersection* of several demographical variables such as race, sex, age, economic status, and geographic location. For example, in [3], we demonstrate that a model for predicting recidivism with an acceptable overall accuracy had an accuracy worse than random guess for the subgroup of Hispanic females, due to inadequate representation. Similar disparities have been highlighted in [5]. This is known as intersectional (aka. subgroup) (un)fairness [6–8, 10].

We must ensure that there are enough entries in the dataset for each subgroup, defined as the intersection of multiple attributes, to prevent population bias and its consequences. We refer to this concept as *coverage* [3]. Specifically, we use "patterns" to represent the intersectional subgroups in the form of attribute value combinations. We require that each pattern have coverage above a specified threshold, τ. For example, {race=Hispanic, gender=female} is a pattern that represents all instances of Hispanic female individuals. Using this, we introduce the notion of *Uncovered Patterns* for identifying the subgroups for which there are not "enough" instances in the dataset. Since there can be many uncovered patterns, we are usually interested in only the most general ones among these. We defined a *Maximal Uncovered Pattern* (MUP) to be an uncovered pattern for which each of its parent patterns is covered by the dataset. For example, assume that {race=Hispanic, gender=female} is a MUP in the COMPAS [1] dataset (as shown in Fig. 1(c)). This means that even though there are enough females and enough Hispanics in the dataset, there are not enough Hispanic females. Also, it is clear that any intersection with a MUP (e.g. Hispanic females under the age of 30) is also uncovered.

In this demonstration, we present MithraCoverage, a system for investigating population bias by identifying MUPs in a given dataset. MithraCoverage leverages the algorithms developed in [3] for MUP detection. As we shall further elaborate in §2, MithraCoverage provides an interactive UI, using which a user can identify a dataset to be investigated and set up the investigation parameters. The system then provides visual information that allows the user to explore through the MUPs and interact with the system.

In the rest of the paper, we first provide our system details, its architecture, implementation, and user interface in §2. Next, in §3, we shall provide our demonstration plan.

2 SYSTEM DETAILS

MithraCoverage is a human-in-the-loop system with a web-based front-end (shown in Fig. 1) and a MUP search engine as the back-end. The web service is built under the CherryPy framework (v18.5) and written in Python 3.7 and the front-end uses the standard technologies including HTML, CSS, JavaScript along with Bootstrap (v4.4) and D3 (v3.5) libraries. The MUP search engine is written in Java 8.

We describe different components in the system front-end and how the user interact with the system in §2.1 and discuss the algorithmic details of the MUP search engine in §2.2.

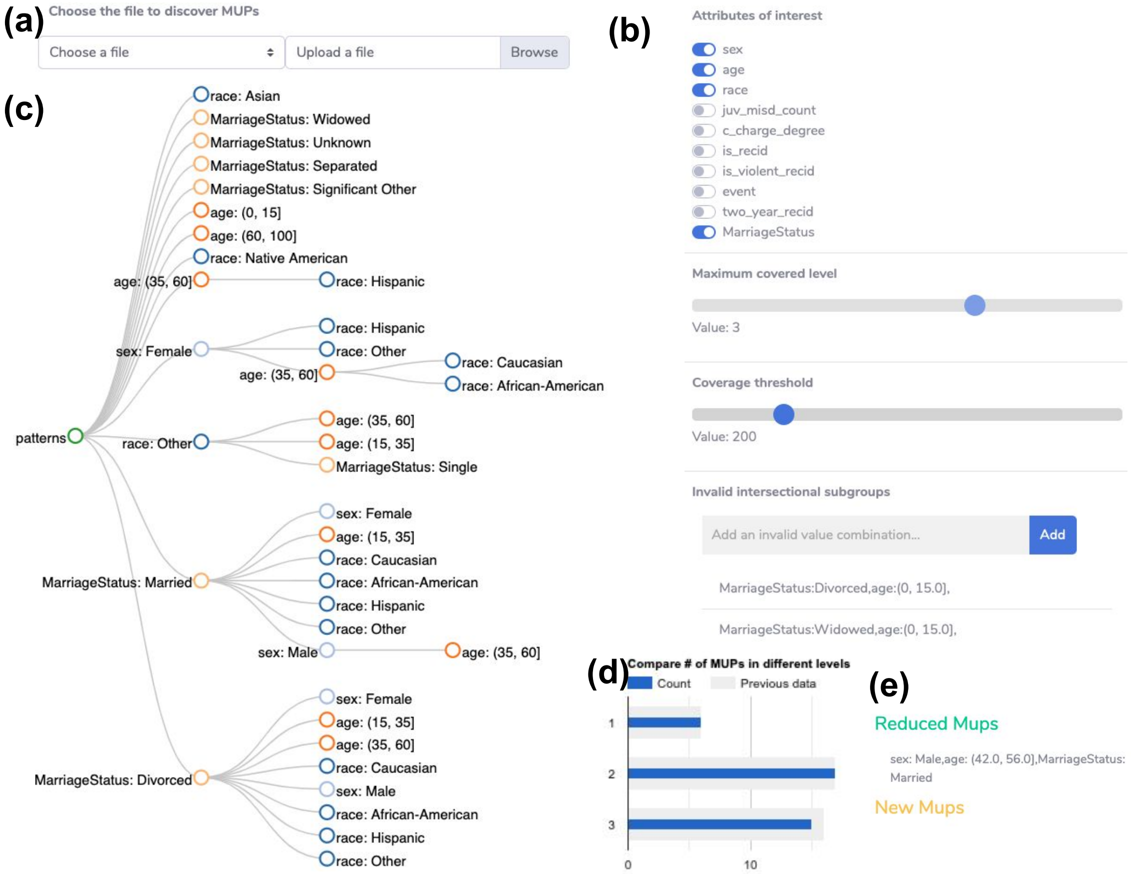
2.1 User Interface

The MithraCoverage front-end user interface consists of four components. The first two components—Data Selection and MUP Search Configuration—are user input and the other two—MUP Chart and Diff Checker—are system output.

Data Selection (Upload) — As shown in Fig. 1(a), the end user selects or uploads a dataset, $\mathcal{D}$, in the form of a csv file, on which she wants to investigate coverage.

MUP Search Configuration — As shown in Fig. 1(b), the user manages the configurations for the following MUP discovery process. As an interactive system, MithraCoverage must ensure its usability. In [3] we show that since the underlying search space is exponential, no polynomial algorithm can guarantee the discovery of all MUPs. As a result, the process of finding all MUPs may become inefficient and, hence, not interactive. There are four parameters the user can tune in MithraCoverage to get meaningful results on time: 1) attributes of interest, 2) coverage threshold, 3) maximum uncovered level, 4) invalid intersectional subgroups.

- **Attributes of interest $\mathcal{A}$.** Not all attributes within a dataset are important for studying coverage. The "attributes of interest" setting lists all columns/attributes in the dataset, and let the user choose a subset, $\mathcal{A}$, where the intersectional fairness need to be concerned. The selected attributes should be categorical. However, if an attribute of interest is continuous, we offer the "bucketization" feature as a preprocessing step for user to easily convert continuous attributes to categorical ones.
- **Coverage threshold τ.** The "coverage threshold" indicates the minimum number of data entries an intersectional subgroup should have in order to qualify as "covered" or "enough". If an intersectional subgroup has at least τ data entries, the pattern for this subgroup will not be returned as a MUP to the end user.
- **Maximum covered level λ.** The "maximum covered level" dictates the maximum number of attributes that their intersection is considered as a subgroup. Intuitively, the more



Choose the file to discover MUPs

Choose a file | Upload a file

Attributes of interest

- ● sex
- ● age
- ● race
- ○ juv_misd_count
- ○ c_charge_degree
- ○ is_recid
- ○ is_violent_recid
- ○ event
- ○ two_year_recid
- ● MarriageStatus

Maximum covered level

Value: 3

Coverage threshold

Value: 200

Invalid intersectional subgroups

Add an invalid value combination...

MarriageStatus:Divorced,age:(0, 15.0],

MarriageStatus:Widowed,age:(0, 15.0],

Reduced Mups

sex: Male,age: (42.0, 56.0],MarriageStatus: Married

New Mups

Figure 1: MithraCoverage **User Interface**

the number of attributes, the less "important" the subgroup is. For example, the group of {race=Hispanic, gender=female, marital status=married, age=20-30} (level = 4) is not as important as {race=Hispanic, gender=female} (level = 2). Thus, it is reasonable to allow the user to set a bound of λ to avoid assessing high-level subgroups and accelerate the MUP search. If a user is interested in exploring lower levels of coverage, she should set this to be lowest level she desires, and the choose what to see by selectively expanding the tree shown in the MUP chart below.

- **Invalid intersections $\mathcal{P}_{-}$.** Although all value combinations are assessed for intersectional fairness, some subgroups may be semantically meaningless and hence should not be returned as the output. Take the subgroup {marital_status=married, age=0-15} as an example. If the legal marriage age is greater than that, as it is in most jurisdictions, returning it as the output is meaningless. "Invalid intersections" is a section of MithraCoverage that allows the user to specify value combinations to exclude from consideration. An "autocomplete" feature is offered in MithraCoverage to facilitate the specification of invalid intersections by end users for any chosen dataset.

MUP Chart — Once the input data and system parameters are configured, MithraCoverage will return the set of MUPs discovered by the search engine as a "MUP Chart", the tree hierarchy shown in Fig. 1(c). Each node in a tree denotes an attribute value, and various attributes can be distinguished by node colors. Each path from the root node to the leaf node represents a discovered MUP pattern combining all the nodes in the path. The tree structure helps to visualize high-dimensional data points (MUPs) in 2D space.

Coverage Enhancement — The user can choose to repair the low coverage issue for certain international subgroups presented in the above MUP chart by collecting and loading additional data points into MithraCoverage. Optionally, the user can click on the "Repair Recomendation" button to view the minimum data record to collect, recommended by MithraCoverage based on the coverage enhancement technique in [3], to resolve the coverage issue at a specific level ℓ (i.e., all valid value combinations of up to ℓ attributes will be covered). Once the user loads additional data points, MithraCoverage also gives the user the ability to compare the impact of additional data on coverage enhancement. That is, besides a new MUP chart, MithraCoverage will present a bar chart as shown in Fig. 1(d) comparing the distribution of MUPs in the new dataset as opposed to the old dataset at all levels. In the bar chart, x-axis is the count of MUPs and y-axis is the level. The blue bars represent the counts of MUPs in the new dataset and the grey bars represents those of the previous dataset. Ideally, when all parameters are fixed, the number of low-level MUPs should be smaller in the new dataset indicating the more severe coverage issues are addressed. Additionally, Diff Checker will highlight both the reduced and new MUPs as shown in Fig. 1(e).

2.2 MUP Search Engine

Once the dataset $\mathcal{D}$, attributes of interest $\mathcal{A}$, coverage threshold τ, maximum coverage level λ and invalid intersections $\mathcal{P}_{-}$ are given, the MUP search engine is supposed to find a complete set of lowest-level MUPs (i.e., intersectional subgroups) within the level λ. The problem itself is proved to be #P-hard. A naïve enumerative search algorithm will quickly becomes intractable from the combinatorial explosion of the problem. We use properties such as *monotonicity* for pruning the search space. Using these properties, we propose three algorithms—PatternBreaker, PatternCombiner, DeepDiver—that have achieved significant speedup than the naïve approach in our experiments. Details of all three algorithms and the proofs are discussed in [3]. Our demo uses DeepDiver as the backbone of the search engine.

3 DEMONSTRATION PLAN

We will demonstrate MithraCoverage using three real-world datasets:

- *COMPAS*[1]*:* ProPublica is a nonprofit organization that produces investigative journalism. They collected and published the COMPAS dataset as part of their investigation into racial bias in criminal risk assessment. The dataset contains demographics, recidivism scores, and criminal offense information for 6,889 individuals.
- *Adult Income Dataset*[2]*:* The dataset contains income information of individuals from 1994 U.S. census. The incomes are split in two classes of ≤ 50K and above 50k. The dataset contains 48842 records over 14 attributes, including `race`, `sex`, `age`, `work-class`, `marital-status`, and `education`.
- *AirBnB*[3]*:* AirBnB is a popular online peer to peer travel marketplace that provides a framework for people to lease or rent short-term lodging. We use a collection of the information of approximately 2 million real properties enlisted in AirBnB. We use this dataset to demonstrate the scalability of MithraCoverage. The dataset provides 41 attributes for each property, out of which 36 are boolean attributes, such as `TV`, `internet`, `washer`, and `dryer`.

We use the first dataset, COMPAS, to demonstrate how a user would interact with MithraCoverage[4]:

1. In the left sidebar, click "Data" button. In the opened box titled "Choose File", select "COMPAS.csv".
2. Click "Configure" in the left side bar to open the configuration box. To configure the MUP search engine, for example, select `sex`, `race`, `age`, and `marital status` as the attributes of interest, let the "maximum covered level" and "coverage threshold" be three and 100, respectively. Aided by the auto-complete feature identify the invalid combinations, such as "Marriage:Divorced,age:(0, 15.0]".
3. Once the configuration is complete, click "MUP" button in the left sidebar'. A tree-structured MUP chart like Figure 1(c) will be presented in the box for the user to overview the intersectional subgroups with coverage issues.
4. Furthermore, the user can click the "Coverage Enhancement" button in the left sidebar and upload a new dataset "COMPAS_additional.csv" as an enhancement for the previous dataset, MithraCoverage will present the bar chart and the MUP diff list as those in Figure 1(d) and (e), besides a new MUP chart, for the user to compare two datasets.

4 ACKNOWLEDGEMENT

This work was supported by NSF Grant No. 1741022 and 1934565. We are grateful to the University of Toronto, Department of Computer Science, and Dr. Nick Koudas for the AirBnB dataset.

REFERENCES

[1] Compas recidivism risk score data and analysis. https://bit.ly/2QzJ0Ci.

[2] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: Risk assessments in criminal sentencing. *ProPublica*, 2016.

[3] A. Asudeh, Z. Jin, and H. Jagadish. Assessing and remedying coverage for a given dataset. In *ICDE*, pages 554–565. IEEE, 2019.

[4] S. Barocas and A. D. Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671, 2016.

[5] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAT**, 2018.

[6] C. Dwork and C. Ilvento. Group fairness under composition, 2018.

[7] J. Foulds, R. Islam, K. Keya, and S. Pan. Bayesian modeling of intersectional fairness: The variance of bias. *CoRR:1811.07255*, 2018.

[8] J. Foulds and S. Pan. An intersectional definition of fairness. *CoRR:1807.08362*, 2018.

[9] A. Hern. Google's solution to accidental algorithmic racism: ban gorillas. The Guardian, 2018.

[10] M. Kearns, S. Neel, A. Roth, and Z. S. Wu. An empirical study of rich subgroup fairness for machine learning. In *FAT**. ACM, 2019.

[11] M. Mulshine. A major flaw in google's algorithm allegedly tagged two black people's faces with the word 'gorillas'. Business Insider, 2015.

[12] A. Olteanu, C. Castillo, F. Diaz, and E. Kiciman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data*, 2:13, 2019.

[13] A. Rose. Are face-detection cameras racist? Time Business, 2010.

[1] www.propublica.org

[2] archive.ics.uci.edu

[3] www.airbnb.com

[4] A video for MithraCoverage can be found at http://bit.ly/2TK5pyj.