



Representation Bias in Data: A Survey on Identification and Resolution Techniques

NIMA SHAHBAZI, University of Illinois Chicago, USA

YIN LIN, University of Michigan, USA

ABOLFAZL ASUDEH, University of Illinois Chicago, USA

H. V. JAGADISH, University of Michigan, USA

Data-driven algorithms are only as good as the data they work with, while datasets, especially social data, often fail to represent minorities adequately. Representation Bias in data can happen due to various reasons, ranging from historical discrimination to selection and sampling biases in the data acquisition and preparation methods. Given that “bias in, bias out,” one cannot expect AI-based solutions to have equitable outcomes for societal applications, without addressing issues such as representation bias. While there has been extensive study of fairness in machine learning models, including several review papers, bias in the data has been less studied. This article reviews the literature on identifying and resolving representation bias as a feature of a dataset, independent of how consumed later. The scope of this survey is bounded to structured (tabular) and unstructured (e.g., image, text, graph) data. It presents taxonomies to categorize the studied techniques based on multiple design dimensions and provides a side-by-side comparison of their properties.

There is still a long way to fully address representation bias issues in data. The authors hope that this survey motivates researchers to approach these challenges in the future by observing existing work within their respective domains.

CCS Concepts: • **Information systems** → **Data management systems**;

Additional Key Words and Phrases: Responsible data science, fairness in machine learning, data equity systems, data-centric AI, AI-ready data

ACM Reference format:

Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and H. V. Jagadish. 2023. Representation Bias in Data: A Survey on Identification and Resolution Techniques. *ACM Comput. Surv.* 55, 13s, Article 293 (July 2023), 39 pages.

<https://doi.org/10.1145/3588433>

1 INTRODUCTION

Data-driven decision-making shapes every corner of human life, from autonomous vehicles to healthcare and even predictive policing and criminal sentencing. A critical question, particularly in applications impacting human beings, is how trustworthy the decision made by the system

This research was supported in part by the National Science Foundation, under grants 2107290, 1741022, 1934565, and 2106176.

Authors' addresses: N. Shahbazi and A. Asudeh, University of Illinois Chicago, 851 S. Morgan St., 11th Floor SEO, Chicago, IL 60607; emails: {nshahb3, asudeh}@uic.edu; Y. Lin and H. V. Jagadish, University of Michigan, 2260 Hayward St. Ann Arbor, MI 48109; emails: {irenelin, jag}@umich.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0360-0300/2023/07-ART293 \$15.00

<https://doi.org/10.1145/3588433>

is. It is easy to see that the accuracy of a data-driven decision depends, first and foremost, on the data used to make it. After all, the system learns the phenomena that data represent. As a first step, we may desire that the data should represent the underlying data distribution from which the production data will be drawn. But that is not enough, since it only tells us about the overall model performance. Although a system may generally perform well in terms of accuracy, it could fail for less populated regions in the data with insufficient representation. These regions may matter, because they frequently represent some minority (sub)population in society. They could also represent cases that may not happen very often but have a relevant impact on the correctness of a critical decision. In short, if data is not representative of a given population, then the outcome of the decision system for that subpopulation may not be trustworthy.

Representation Bias happens when the training data under-represents (and subsequently fails to generalize well) some parts of the target population [115]. Data representation bias can originate from how (and from where) the data was originally collected or be caused by the biases introduced after collection, either historically, cognitively, or statistically. Representation bias can happen due to selection bias, i.e., when the sampling method only reaches a portion of the population or the population of interest has changed or is distinct from the population used during model training. For example, a survey to measure the illegal drug use of teenagers could be biased if it only includes high school students and ignores home-schooled students or dropouts. Another potential reason is the skewness of the underlying distribution. Suppose the target population for a particular medical dataset is adults aged 18–60. There are minority groups within this population: For example, pregnant people may make up only 5% of the target population. Even with perfect sampling and an identical population, the model is prone to be less robust for the group of pregnant people because it has fewer data points to learn from [115]. Furthermore, even if we carefully arrange for uniform sampling by age, we may find that sampling is non-uniform for pregnant people. For example, there may be proportionately fewer pregnant people over 40. If some group is a minority in the underlying distribution, then even random sampling will not help the under-representation issue for this group.

Representation bias is almost always guaranteed without a systematic approach to data collection. For example, in a survey data collection, a crucial step is to identify all the sub-populations in the underlying distribution based on the desired demographic information and ensure that the survey reaches all of them while enough samples are collected from each. However, the problem is that data scientists usually do not have any control over the data collection process, resulting in the utilization of “found data” in most data-driven decision-making systems. Therefore, with no guarantee on the aforementioned steps in the data collection process, the found data is most likely a biased sample.

Representation bias in data is not a new problem and has been a known issue in data mining, database management, and statistics communities. There is a rich line of work on the problem of discovering interesting patterns, regularities, or finding empty space in the data that is a parallel and relatively similar problem to identifying representation bias in datasets [40, 74, 78, 79]. However, with the emergence of responsible data science and trustworthy AI, this problem has been addressed with greater vigor and from a brand new perspective in recent years. This survey discusses techniques for identifying and resolving representation bias in datasets, introducing taxonomies to classify these techniques based on multiple dimensions. Note that while the literature on algorithmic fairness is primarily concerned with promoting fairness in **machine learning (ML) models**, bias is sought to be addressed in the *datasets*, regardless of how the data is ultimately consumed.

We start the article by presenting a big-picture overview of the *fairness literature* in Section 2. This will help us specify the scope of this survey w.r.t. fairness approaches and existing surveys.

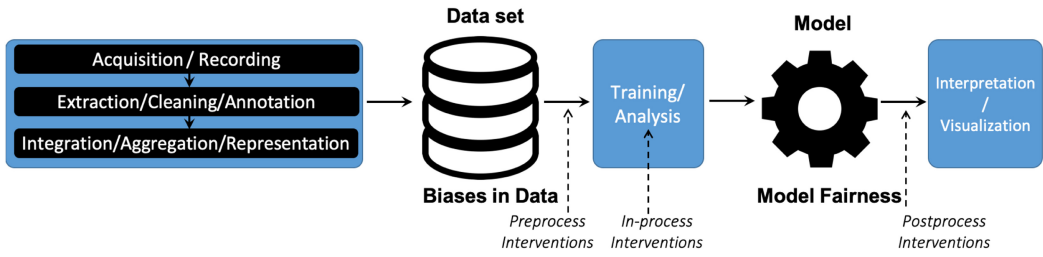


Fig. 1. Illustration of bias and fairness in data analytics pipeline (the pipeline is adapted from Reference [60]).

Next, in Section 3, we zoom in on the notion of *representation bias*, explaining the reasons that give rise to it and presenting techniques for measuring representation bias. In Section 4, we propose a taxonomy to categorize different approaches to identify and resolve representation bias in *structured data* based on factors such as objectives and capabilities. Following our taxonomy's guidelines, we investigate each work's details, explain its novelty, and discuss its pros and cons. In Section 5, we review the techniques for identifying and resolving representation bias in *unstructured data* such as images, text, speech, and graphs. Finally, in Section 6, we present an overview of the reviewed works and conclude the survey by discussing aspects that have been less noticed in the existing lines of work and propose some possible directions for the researchers to investigate.

2 AN OVERVIEW OF FAIRNESS LITERATURE

As AI replaces human beings in various critical fields, the topic of fairness among the affected population becomes more crucial. In recent years, the general topic of fairness has drawn sizable attention from different communities, specifically in the ML field. Many surveys [14, 25, 83, 98, 111] and tutorials have been published on the related topics and even the conference ACM FAccT¹ has been dedicated to this topic. Before focusing on representation bias in a dataset, it is beneficial to review the big picture of fairness literature, including the definitions and techniques to achieve fairness.²

While there is no clear agreement on the definitions of fairness, it can be viewed from three perspectives [15]: *individual fairness*, *group fairness*, *subgroup fairness*. Individual fairness is the most granular notion of fairness, requiring similar outcomes for similar individuals. Group fairness definitions are satisfied if a model has equal or similar performance on different groups w.r.t. the associated fairness measures. Subgroup fairness falls between individual and group fairness and is satisfied when groups are defined over the intersection of values of multiple sensitive attributes. Fairness can be considered by ML models at different stages of the data analysis pipeline, shown in Figure 1. As highlighted in the figure, the intervention strategies to achieve model fairness fall under three categories: *Pre-process*, *In-process*, and *Post-process* interventions. The main idea of the pre-processing category of techniques is to modify the data before feeding it into the ML algorithms. In-process methods mainly reinforce fairness by inducing constraints or adding regularization terms to the objective function of the learning algorithm. Post-process methods manipulate the results of a classifier to promote fairness among different groups. Given this context, we will specify the scope of this survey.

¹<https://faccconference.org/>.

²For an extensive discussion on the fairness literature, please see the online version of this survey.

2.1 Scope of the Survey

Having discussed fairness definitions and the intervention approaches to achieve it, let us look at the pipeline of data analytics in Figure 1 again. The model is often considered “the product” of the pipeline. Indeed, works on fairness focus on the model. However, the *dataset* is also a product, of possible interest in its own right, in addition to its influence on model fairness. Given a dataset to train a model, fairness intervention techniques aim to build a model based on some fairness criteria. However, this survey focuses on the other product of the pipeline, i.e., datasets, studying *bias as a feature of a dataset*, independent of how it is later consumed. In particular, the scope of the studies reviewed in this survey is bounded to *representation bias in structured (tabular) and unstructured (image, graph, text, speech) data*.

3 AN OVERVIEW OF REPRESENTATION BIAS

With the abundance of data collected from a wide range of contexts, we are transitioning from decision-making based on intuition and anecdotal observations to decision-making based on the data. Data-driven decision-making has great potential, and success stories abound. But there are also failures, usually because the larger volume can make it easier to hide many problems. It is said that every decision is only as good as the data used to make it [16]. One of the most important, aspects of data quality is being representative of all the possible subgroups influenced by that decision [47]. This representativeness originates from how the data has been collected. With a prospective data collection approach, such as through a survey or a scientific experiment, data scientists may be able to specify requirements like representation in data. However, more often than not, data, now known as found data, is collected independently in a process that data scientists have limited or no control over. Besides, it is important to note that while data must follow the actual production distribution, this is not sufficient for the development of representative data. The data must include enough examples from “less popular regions” of data space if these regions are to be handled well by the system.

In today’s data-driven world, **Automated Decision Systems (ADS)** are widely used in society, ranging from fire prevention by predicting high-risk buildings to recruiting automation by screening for competitive candidates. However, historical data used for decision-making might not be objective; it could inherit historical biases in the algorithm design. For the responsible development of ADS, it is essential to analyze the representation to avoid the potential risks of injustice. For example, an attempt of the Boston government [45] using a system to assign students to schools near their residential areas was found problematic, as it ignored the fact that top schools are typically less common in underprivileged districts. For systems that rely on machine learning algorithms, without a careful inspection of the training data quality, under-representation of minority groups may cause discrimination in the prediction results [8, 28, 47, 77]. For example, StyleGAN [66], one famous algorithm for auto-generating eerily realistic human faces, is also producing white faces more frequently than faces of people of color. The problem appears inherited from the training datasets, which default to white features. As a result, recent research has started to explore the relationship between machine learning bias and the inadequate sample sizes [28, 93]. Representation bias is also a crucial problem in critical domains, such as health care. First, there are group-specific patterns in the healthcare data. For example, many diseases are correlated with demographic factors like race, gender, and so on. Ashkenazi Jewish women are known to have a higher risk of breast cancers [41]; the likelihood of many diseases, including obesity, hypertension, diabetes, and high total cholesterol, also varies across racial/ethnic groups [1]. Therefore, the medical datasets’ diversity, especially demographical diversity, is vital when further using the collected data. Besides, as health data are usually sensitive, patients’ willingness to share the data might vary [33].

As a result, ensuring the representativeness of the collected data is essential to avoid inaccurate or biased results in the downstream usage of the data.

3.1 Reasons for Representation Bias

Bias has been studied in the statistical community for a long time [91] but social data, increasingly used for policy decision-making and by social scientists and digital humanities scholars, presents a set of different challenges [15, 16, 94]. At a high level, bias in social data means certain sub-populations in data are more heavily weighted or represented due to systematic favoritism. It is a deviation from expectation in data and is recognized as a subtle error that sometimes goes unnoticed, causing skewed outcomes, low accuracy levels, and analytical errors. These biases are sometimes introduced to the data due to cognitive biases [55, 56] in human reporting or flawed data collection or preprocessing. We refer the reader to References [53, 94] for more information about the general topic of biases in social data, the origins, and various types.

The center point of this survey, representation bias, happens for a variety of reasons with no consensus on an exact set of grounds. With that in mind, we seek the origins of representation bias in one or more of the following:

Historical Bias. Historical bias is “the already existing bias due to the socio-technical issues in the world” [83]. An example of historical bias can be found in Google’s image search results. Searching for the term “CEO United States,” the results are dominated by images of male CEOs and show fewer female CEO images. This is because only 8.1% of Fortune 500 CEOs are women, causing the search results to be biased towards male CEOs. This problem has previously been shown for a variety of job titles, such as “CEO” in Reference [72], and Google had alleged to have resolved it. These search results are indeed reflecting reality. However, whether the search algorithms should mirror this reality or not may depend on the application and is another issue to consider.

Underlying Distribution Skew. The underlying distribution that data is collected from may lack an equal ratio or sufficient representation for all of its subpopulations. In such cases, the underlying distribution is inherently skewed, and there are no discriminatory motives behind it. For example, according to the US Census Bureau [2], around 7% of the US population is of Asian descent, while 75% of the population is White. Collecting a uniform sample from the US society, the Asian community is considered a minority in the outcome sample and naturally less represented. However, this is a reflection of the underlying distribution that the data has been collected from. This reflection of reality may lead to discrimination against this subpopulation in some applications.

Sampling/selection/self-selection Bias. Selection bias is introduced to the data when one fails to ensure proper randomization in selecting people, groups, or tuples of data for analysis. *Sampling bias* happens on account of a non-random sampling of a population, causing some (sub) populations to be less likely to be sampled. Note that selection bias is a cause for sampling bias, since having selection bias, the collected samples may not represent a random sampling of a population. *Self-selection bias*, however, happens when only a subset of a selection population chooses to participate in an experiment. This bias occurs when the intention of the participants whether to participate in the research or not creates abnormal or undesirable conditions. Although selection bias, sampling bias, and self-selection bias are sometimes used interchangeably, it is important to differentiate between them. Let us clarify this distinction using an example. Consider a researcher who would like to conduct a survey in Chicago, mailing ballots to selected respondents. Now, if the respondents are only selected from some regions (e.g., near downtown), hence failing to ensure a random representation of different populations in the city, then this is an example of *selection*

bias. Suppose there is no bias in the selection of respondents. However, only a small portion of the invited respondents decide to take the survey and mail the forms back. This can cause the *self-selection bias*. To see how, let us consider the famous example where the survey question is “Do you like responding to surveys?” with two possible options: (1) Yes, I love responding to surveys (2) No, I toss them in the trash. Now, suppose only 10% of the respondents opted to take the survey and the collected results show 99% favored option 1. The result is indeed invalid, as the other 90% who decided not to take the survey would likely have selected option 2! Now, independent of how the survey was taken, if the collected samples are not random over Chicago’s population, then it is an instance of *sampling bias*.

3.2 Measuring Representation Bias

In this section, we discuss the measures that have been proposed to evaluate representation bias in data.

3.2.1 Representation Rate. Representation rate is a metric defined in Reference [26] to identify representation bias w.r.t. the base rates. Base rate, also known as “prior probability,” refers to the class probability unconditioned on any observation. In the existing works [69, 109], an equal base rate is defined as having an equal number of objects for different subgroups in the dataset. In other words, the objects in the selected set should have an equal chance of belonging to each subgroup. Consider dataset \mathcal{D} with n tuples and let n_i be the number of tuples belonging to subgroup i . That is, for all possible subgroups i, j in \mathcal{D} , they are represented if $n_i = n_j$.

Next, we present the definition of the representation rate. Consider dataset \mathcal{D} from discrete domain $\Omega := \Omega_1 \times \dots \times \Omega_d = \{0, 1\}^d$ where d is the number of dimensions of the dataset. For a threshold $\tau \in (0, 1]$, dataset \mathcal{D} following the distribution $p : \Omega \rightarrow [0, 1]$ is said to have representation rate of τ with respect to a sensitive attribute ℓ if for all $z_i, z_j \in \Omega_\ell$, we have $\frac{p[Z=z_i]}{p[Z=z_j]} \geq \tau$. That is, for all possible subgroups i, j , we have $\frac{n_i}{n_j} \geq \tau$. The closer τ is to zero, the more biased \mathcal{D} is. Representation rate might be hard to achieve. That is because, in practice, it rarely happens that all subgroups have (almost) the same number of objects.

3.2.2 Data Coverage. The notion of data coverage has been studied across different settings in References [5, 7–9, 62, 77, 86, 116] as a metric to measure representation bias. At a high level, coverage is referred to as having enough similar entries for each object in a dataset. For a better understanding, let us go over a definition for the generalized notion of coverage. Consider a dataset \mathcal{D} with n tuples, each consisting of d attributes $X = \{x_1, x_2, \dots, x_d\}$. Attribute values may be non-ordinal categorical (e.g., race) or continuous-valued (e.g., age). Ordinal attribute values are normalized to lie in the range $[0, 1]$, with values drawn from the set of rational or real numbers. For every tuple $t \in \mathcal{D}$, $t[i]$ shows the value of t on attribute $x_i \in X$. In practice, the data scientist may be interested in studying coverage over a subset of attributes, called “*attributes of interest*.” Examples of attributes of interest are *gender*, *race*, *salary*, and so on. Subsequently, X is assumed to be the set of attributes of interest. The dataset also contains target attributes $Y = \{y_1, \dots, y_d\}$ that may or may not be considered for the coverage problem.

Given a query point $q \in [0, 1]^d$, where $q[i]$ shows the value of q with regard to $x_i \in X$, q is not covered by the dataset \mathcal{D} , if there are not “enough” data points in \mathcal{D} that are representative of q . To generalize the notion of coverage, let us define $\mathcal{G}(q)$ as the group of tuples that would represent q . For example, suppose $X = \{\text{gender}\}$ and q has gender=female. Then, the set of female individuals represents q . Let $\mathcal{G}_{\mathcal{D}}(q) = \mathcal{G}(q) \cap \mathcal{D}$. That is, $\mathcal{G}_{\mathcal{D}}(q)$ are the set of tuples in \mathcal{D} that represent q . Using this notation, coverage of q is defined as the size of $\mathcal{G}_{\mathcal{D}}(q)$. That is, $\text{cov}(q, \mathcal{D}) = |\mathcal{G}_{\mathcal{D}}(q)|$. Given a coverage threshold value k , q is covered if and only if $\text{cov}(q, \mathcal{D}) > k$. The *uncovered region* in a dataset is the collection of tuples that are not covered by it.

It is important to have a high enough coverage for all meaningful sub-populations in data regardless of the data space to make sure they are adequately represented. We would also like to emphasize the necessity of *human-in-the-loop* to ignore semantically incorrect sub-populations, e.g., {gender=male, isPregnant = {True}}. Coverage thresholds are expected as an input to the problem and are supposed to be determined through statistical analyses as they are application-specific and vary by context. By borrowing the concept from statistics and central limit theorem, the rule of thumb suggests the number of representatives be around 30, or as Reference [112] suggests, for each “minority subpopulation” a minimum of 20 to 50 samples is necessary.

3.2.3 Representation Rate vs. Data Coverage. Having discussed representation rate and data coverage, let us further compare these two measures with an example. Consider a dataset \mathcal{D} with 1,000 tuples each having an attribute {gender} with values {male, female}. To satisfy representation rate requirements, the male and female groups should have close counts *relatively to each other*. For example, using the threshold $\tau = 0.8$, the ratio of females-males (assuming that females are the minorities) should be at least 80%. In other words, given that the dataset size is 1k, the dataset should contain at least 445 females. However, data coverage requires a minimum count for each of the groups *independent from the counts on other groups*. So, for coverage threshold value $k = 100$, each of the male and female groups should at least have 100 tuples to be covered. Finally, comparing the two measures, it is evident that the representation rate provides stronger guarantees of resolving issues w.r.t. representation bias in downstream tasks, however, it is more restrictive and harder to achieve compared to the data coverage. In particular, when the underlying distribution is skewed (as explained in Section 3.1), it is not possible to both follow the underlying distribution and fully satisfy the representation rate.

A connection between the fairness measures and representation bias has been made to prove fairness impossibility theorems. In particular, Kleinberg et al. [69] prove when there is an unequal base rate in data (i.e., representation rate is less than one), it is not possible to satisfy different fairness measures. For example, it is not possible to achieve both Equalized Odds and Predictive Parity at the same time.

3.3 Representation Bias Harms

Before starting the discussion on representation bias identification, we would like to underscore that although representation bias is important, it does not necessarily imply poor and groundless decision-making of the system. For example, in a classification setting, having representation bias on continuous attributes in regions far from the *ground-truth* decision boundary is likely to be immaterial, since those points may not contribute to refining the boundary. Similarly, in a regression setting, in regions of the training data where the fluctuation of the target value is not much, representation bias is much less crucial than in regions with a higher fluctuation. In general, it is safe to say that representation bias is problematic in the regions where the model behind the decision system fails to interpolate adequately based on the current data sample.

To further verify this, let us consider the following experiments, adapted from Asudeh et al. [9]. First, consider a binary classification task to label a query point on the x-y plane as belonging to the body of a cat image or the background (Figure 2(a)). The training data is generated by randomly sampling from the image, labeling each sample point as +1 if inside the cat’s body and -1 otherwise. Next, we intentionally remove the sample points in the training data that belong to the patch highlighted in Figure 2(b) to make it under-represented. Using the training data and trying multiple classification models, while the overall performance of the classifiers is high, they all fail to work for the under-represented region. In particular, while the overall false-negative rate was less than 5%, it was as high as 54% for the under-represented region. Relying on the training

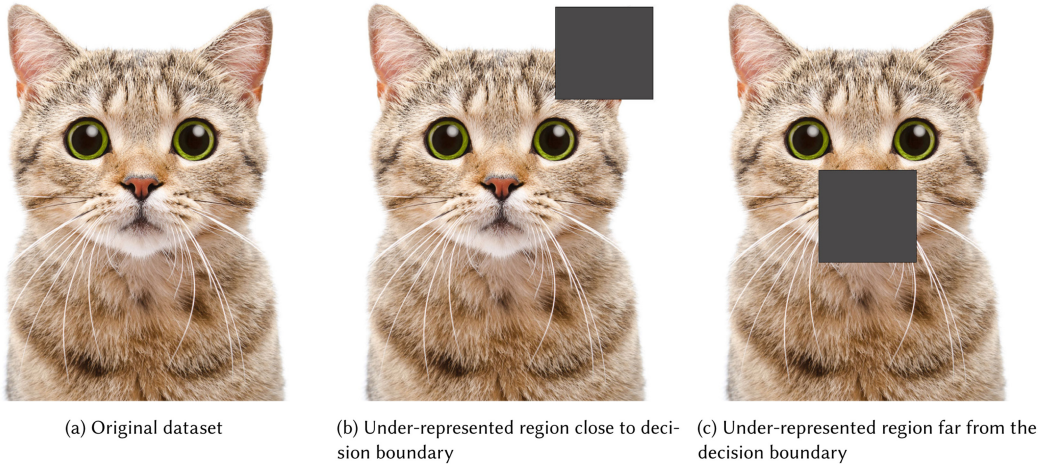


Fig. 2. Classification task: whether a query point is inside/outside the cat body. Illustration of classifier's performance for different under-represented regions.

data, the models create the decision boundary by connecting the two edges of the cat's body, missing its ear. As a result, the query points that belong to the ear are misclassified as background, resulting in a high false-negative rate.

Next, we repeat the experiment, but this time, we remove the sample points belonging to the patch shown in Figure 2(c). The performance difference of the models between the overall image and the under-represented region is relatively small (around 4%), and the model performs well for the under-represented region. Looking at the training data, the patch does not contribute to defining the decision boundary in this specific classification task and, therefore, has minimal impact on the model's performance.

4 REPRESENTATION BIAS IN STRUCTURED DATA

Structured data (a.k.a. tabular data) is the most common type of data available in the real world. Databases are built upon the concept of organizing data in a structured manner to facilitate tasks such as storage, querying, representation, and so on. Representation bias in structured data has extensively been studied, and various techniques for the related problems have been proposed. This section discusses the literature on identifying and mitigating representation bias in structured data. For each dimension, we will go through a detailed description of the research works and discuss their novelties. Figure 3 depicts the taxonomy we propose for the structured data to categorize different techniques based on their objectives, capabilities, and assumptions.

4.1 Running Example

We use the Adult Income Dataset [76] to present running examples to better clarify the reviewed techniques. The Adult Income Data set is used to predict whether individual income exceeds \$50k/yr based on the census data.

Consider a projection of the Adult Income Dataset, shown in Table 1, with six attributes $\mathcal{A} = \{\text{gender}, \text{race}, \text{marital-status}, \text{age}, \text{hours-per-week}, \text{years-experience}\}$, among which $\{\text{gender}, \text{race}, \text{marital-status}\}$ are non-ordinal categorical and $\{\text{age}, \text{hours-per-week}, \text{years-experience}\}$ are continuous-valued. The data domain for the categorical attributes are $\text{gender} = \{\text{male}, \text{female}\}$, $\text{race} = \{\text{White}, \text{Black}, \text{Asian}, \text{Hispanic}\}$, $\text{marital-status} = \{\text{single}, \text{married}\}$. Any attributes in \mathcal{A}

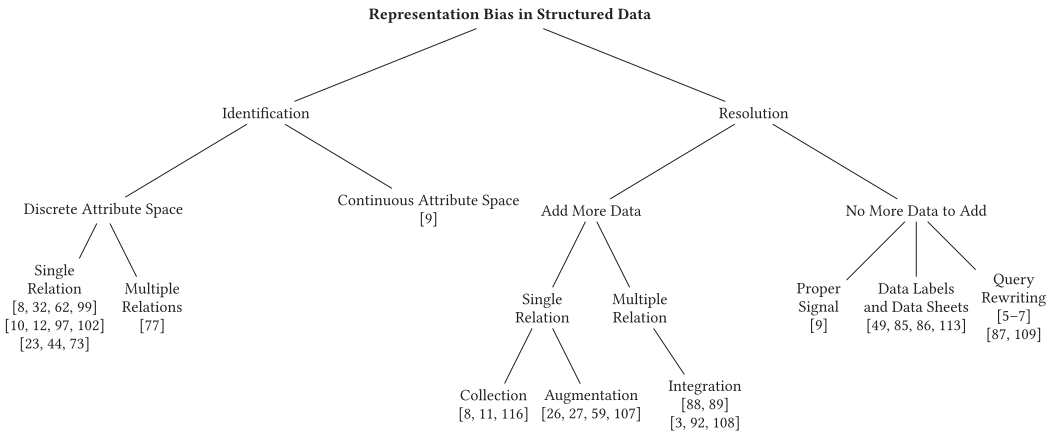


Fig. 3. Classification of techniques on identifying and resolving representation bias in structured data.

Table 1. A Toy Illustration of the Running Example (the Adult Income Dataset)

id	gender	race	marital-status	age	hr/week	yrs-exp	above-50k
1	male	white	single	21	40	3	0
2	female	white	single	28	38	5	0
3	male	white	married	35	45	10	1
4	male	black	single	30	40	8	0
...							

can be considered sensitive attributes. The dataset also contains binary ground-truth $Y = \{1, 0\}$ representing whether an individual makes greater than \$50k annually or not.

4.2 Identification of Representation Bias

In this section, we study the works focused on identifying representation bias in structured data. Depending on the type of the attributes of interest, we categorize the techniques into two classes based on whether they target the problem for *discrete* (non-ordinal; e.g., race, gender) or *continuous* (ordinal; e.g., age) attributes. The attributes of interest considered for representation bias often include sensitive attributes (a.k.a. protected attributes) such as race and gender but are not necessarily limited to them.

4.2.1 Discrete Attribute Space. Let us begin with cases where attributes for identifying representation bias are categorical. To better observe representation bias in such cases, let us consider the following example:

Example 1 (Representation Bias in Discrete Attribute Space). Consider the running example dataset (Table 1) described in Section 4.1. Suppose the categorical attributes {race, gender, marital-status} are used for representation bias identification. Each conjunction of attribute-value assignment for a subset of attributes specifies a subgroup such as {race=black \wedge gender=female}. If there are not enough tuples in the dataset matching a specific subgroup, then it may not be a suitable dataset on which to train a system to make a decision for that group.

The existing work has evaluated representation bias in discrete space using the discrete notion of coverage measure and representation rate. Many critical research fields have targeted the problem of identifying representation bias from different perspectives. For example, in machine learning

it is important to identify under-represented subgroups in the data used to build the models, as they are at a higher risk of experiencing unfairness in downstream data-driven algorithms [8, 62]. Another closely related problem in machine learning is model validation by finding problematic regions in data that the model will perform poorly [32, 97, 102, 116].

Depending on whether the data is single or multiple related, in the following, we will study the techniques for identifying representation bias in discrete structured data.

Single Relation. The majority of the existing works focus on studying representation bias for datasets that populate data in just a *single* table.

We begin with Reference [8], which identifies representation bias in discrete space using the discrete notion of coverage measure. For cases where attributes of interest are non-ordinal categorical, coverage is defined as having “enough” entries in the dataset matching a particular pattern. A *pattern* is a string that specifies a subgroup (e.g., $\text{gender}=\text{male} \wedge \text{race}=\text{white}$) that matches possible values over a subset of attributes of interest. Coverage is usually discussed for groups given by the conjunction of attribute-value assignments. A constant value is considered as the threshold for coverage, meaning that a minimum number of entries equal to the threshold value should exist from a subpopulation to be covered. In discrete datasets, there are multiple attributes each having multiple possible values that form a combinatorial number of possible patterns. Since patterns are the combination of some or all attributes-values, they can have multiple children and parents. A pattern P_1 is the parent of pattern P_2 , if P_1 can be obtained by replacing one of the deterministic elements in P_2 with X . Deterministic elements in a pattern have a specified value, while non-deterministic elements are indicated by X . As a simple example, consider a pattern defined over a single binary attribute *gender* with domain $\{\text{male}, \text{female}\}$. Pattern P_1 : ($\text{gender}=X$) is the parent to either of patterns P_2 : ($\text{gender}=\text{male}$) or P_3 : ($\text{gender}=\text{female}$). Equivalently, patterns P_2 and P_3 are the children of P_1 . Depending on the size and skew in datasets, the coverage of patterns could be different and Asudeh et al. try to identify patterns that do not have sufficient coverage in an efficient way. If a pattern is uncovered, then all of its children are also uncovered. This suggests that uncovered patterns should be identified in a way that is not dominated by more general ones; for example, if patterns P_1 : ($\text{gender}=X$) and P_2 : ($\text{gender}=\text{male}$) are both known to be uncovered, then P_1 is said to dominate P_2 if P_1 is the parent of P_2 . Uncovered patterns that do not have uncovered parents are referred to as **maximal uncovered patterns (MUPs)**. Therefore, the problem of identifying representation bias using the discrete notion of coverage is defined as follows: Given a dataset \mathcal{D} defined over d attributes with cardinalities c , as well as the coverage threshold τ , try to find all MUPs.

No polynomial time algorithm can guarantee the enumeration of the entire MUPs, however, several algorithms inspired by set enumeration and the Apriori algorithm for association rule mining are proposed to efficiently address this problem. In this regard, Asudeh et al. introduce *Pattern Graph* data structure that exploits the relationship between patterns to do less work than computing all uncovered patterns by removing the non-maximal ones. The parent-child relationship between the patterns is represented in a graph that can be used to find better algorithms. *Pattern-Breaker* starts from the top of the graph where the general patterns are and moves down by breaking each pattern into more specific ones. If a pattern is uncovered, then all of its descendants are also uncovered and they can not be an MUP, even if they have a parent that is covered. Therefore, this subgraph of the pattern graph can be pruned. The issue with *Pattern-Breaker* is that it explores the covered regions of the pattern graph and for the cases where there are a few uncovered patterns, it has to explore a large portion of the exponential-size graph. To tackle this, *Pattern-Combiner* algorithm is proposed that performs a bottom-up traversal of the pattern graph.

It uses an observation that the coverage of a node at the level of the pattern graph can be computed as the sum of the coverage values of its children.

Example 2 (Pattern-Combiner (Asudeh et al. 2019)). Consider the subgroup $\text{race}=\text{Asian AND gender}=\text{female}$ in Example 1, this data pattern is in the bottom layer of the Pattern Graph, as it contains no unspecified values. It has no children and three parent data patterns: $(\text{race} = X \wedge \text{gender} = \text{female})$, $(\text{race} = \text{Asian} \wedge \text{gender} = X)$, and $(\text{race} = X \wedge \text{gender} = X)$. If we find $(\text{race} = \text{Asian} \wedge \text{gender} = \text{female})$ has enough coverage, then all its parents are covered. Pattern-Combiner visits the data patterns in the Pattern Graph in a bottom-up manner, and once we find the covered pattern, we can get the coverage of its parents.

The problem with *Pattern-Combiner* is that it traverses over the uncovered nodes first and therefore, it will not perform well for the cases in that most of the nodes in the graph are uncovered. In fact, for the cases where most of the MUPs are placed in the middle of the graph, both *Pattern-Breaker* and *Pattern-Combiner* will not be efficient, as they should traverse half of the graph. Therefore, they propose *Deep-Diver*, a search algorithm based on Depth-First-Search that quickly finds the MUPs, and use them to limit the search space by pruning the nodes both dominating and dominated by the discovered MUPs.

Jin et al. [62], design a system on top of the methods and algorithms proposed in Reference [8] to investigate representation bias over the intersection of multiple attributes using the notion of coverage.

The next work by Chung et al. [32] proposes SliceFinder³ as a solution to address a similar problem to identifying representation bias in data. They try to determine if a model under-performs on some particular parts of data (referred to as a data *slice*), since the overall model performance can fail to reflect that of smaller data slices. A slice is a conjunction of attribute-value pairs (similar to patterns in Reference [8]) and is considered problematic if the classification loss function takes very different values between the slice and the rest of the data. Enumeration of all possible slices is not practical, and searching for the most under-performing slices can be deceptive, since model performance over smaller slices can be noisy or they may be too small to have a considerable impact on the quality of the model. The goal is to identify the top- k largest and most problematic slices for which the model does not perform well. Finding the most problematic slices requires a balance between the significance of the difference in loss and the magnitude of the slice. To do so, the disparity between the loss of a slice and its counterpart is calculated using a loss function like *logarithmic loss* such that the difference is always non-negative (slice has a higher loss than its counterpart). To determine if the difference is significant, Chung et al. suggest treating each slice as a hypothesis and performing two tests to determine (1) if the loss disparity is *statistically significant* (not observed by chance) and (2) whether the *effect size* of the disparity is large enough (how problematic the slice is). Therefore, they find a handful of the largest problematic slices, by taking all problematic slices with an effect size larger than a threshold and ranking them by size (number of entries). To search for problematic slices, Chung et al. propose three algorithms including a baseline. First, they propose the *Decision Tree Training* method, in which they train a decision tree to partition examples into slices defined by the tree. To find the k -problematic slices, they perform a Breadth-First-Search on the decision tree in which slices in each level are sorted based on an increasing number of literals, decreasing slice size, and decreasing effect size and filtered whether they are statistically significant and have large enough effect-size. The advantage

³Note that, unlike previous works, this work (as well as References [10, 11, 44, 97, 99, 102], explained later) is model-aware. While this assumption may place these works in the scope of fairness-related literature, due to their data-centric approaches, we include them in our survey.

of using the decision tree approach is its natural interpretability and the fact that it needs to be expanded a few levels to find the top- k problematic slices. Conversely, a decision tree is optimized for classification results and may not find all problematic slices. Besides, in cases of overlapping data slices, the decision tree will find at most one of them. To overcome the aforementioned problems, they propose the *Lattice Searching* algorithm, in which slices form a lattice and problematic slices can overlap. Lattice searching follows the same procedure as the decision tree training algorithm to search for the problematic slices. Lattice search can be more expensive than the decision tree training approach and cannot address the scalability issue of searching over the exponential size of data slices, therefore, they suggest employing parallelization and sampling techniques. To better clarify how the lattice search algorithm works, let us look into an example:

Example 3 (Lattice Search (Chung et al. 2019)). Consider dataset \mathcal{D} described in Section 4.1. For simplicity, suppose that we are interested in top-2 largest slices only w.r.t. to gender and marital-status attributes, and the effect size threshold is T . Initially, priority queue Q includes the entire data as a slice. This slice does not have the required effect size and thus is expanded into slices gender=male, gender=female, marital-status=single, and marital-status=married that are inserted in the queue. Next, suppose that gender=female slice has the minimum effect size T and is therefore dequeued and added to the top-2 results. With none of the remaining slices having an effect size T , the largest remaining slice (supposedly marital-status=single) is expanded. Suppose marital-status=single \wedge gender=male has the minimum effect size T , then it is added to the top-2 results and the algorithm stops. Note that marital-status=single \wedge gender=female is already considered, as it is a subset of gender=female slice.

Next work, SliceLine [102], expands on the idea of the previous work [32] for exact slice enumeration to find real top- k problematic data slices. This is due to the fact that none of the methods introduced in Reference [32] are able to find the real top- k problematic slices and this uncertainty creates trust concerns. Utilizing frequent itemset mining algorithms and monotonicity for effective pruning, they present a sparse linear algebra implementation of slice enumeration that is efficient in practice. To do so, a scoring function is devised that linearizes the errors and sizes by involving the ratio of average slice error to average overall error and deducting the ratio of overall size to slice size, while weighting these segments by the user parameter α . Using this scoring function, all slices with a score larger than zero are slices of interest and will be returned in descending order of their score. They also propose upper bounds for the scoring function based on which the search lattice can be effectively pruned.

Pradhan et al. [99] propose a related approach to Reference [32] to identify the patterns in the data that are responsible for bias from a causal perspective. Using the same notion of pattern as Reference [8], they use interventions to measure the effect of patterns in data that significantly promote bias. To do so, they remove a subset of data that is assumed to be the root of bias and evaluate whether a classifier built on the remaining data is less discriminatory. The bias of each pattern is evaluated with the *interestingness* measure. Given a fairness metric \mathcal{F} , dataset \mathcal{D} , and pattern p , interestingness of pattern p is defined as $\frac{\mathcal{F}_{\mathcal{D}} - \mathcal{F}_{\mathcal{D}/p}}{Sup(p)}$ where $\mathcal{F}_{\mathcal{D}}$ is the bias of a classifier trained on \mathcal{D} , $\mathcal{F}_{\mathcal{D}/p}$ is the bias of the classifier trained on intervened \mathcal{D} (by removing p from \mathcal{D}), and $Sup(p)$ is the fraction of data points that satisfy pattern p . To find the top- k patterns causing the most bias, Pradhan et al. utilize a similar bottom-up approach to the lattice-based search that we saw in References [8, 32].

Azzalini et al. [10, 12] propose yet another approach to detect representation bias in the data based on **conditional functional dependencies (CFDs)**. CFDs are conditional dependencies that apply to only a subset of tuples specified with a condition. They use techniques proposed in Reference [24] to explore the CFDs and filter out all of the CFDs that do not have at least one sensitive

attribute and target variable or some of the present attributes are not assigned a constant value. Among the remaining CFDs, they calculate the difference in confidence without and with the sensitive attribute on the left-hand side. The confidence value indicates how often the CFD has been true. A positive confidence difference is indicative of bias toward the sensitive group on the left-hand side of the CFD. Finally, they rank the CFDs w.r.t. multiple criteria of interest such as support-based (number of tuples affected by the bias in CFD), difference-based (largest impact of the protected attribute on the right-hand-side), and mean-based (balance between the two prior criteria).

Pastor et al. [97] propose the notion of *divergence* to estimate different classification behavior in subgroups compared to the overall dataset. Divergence measures the difference in statistics such as false-positive rate and false-negative rate between a subgroup and the entire dataset. However similar to Reference [32], to recognize the problematic subgroups, they only consider the most frequent patterns with a size larger than a threshold and discard smaller subgroups. Once subgroups with high divergence are recognized, they check whether they are statistically significant or not due to fluctuations caused by the finite size of the dataset. Next, using the notion of Shapley value [106], they investigate which attributes in each problematic subgroup are contributing the most to the local and global divergence. In this work, Shapley values measure the contribution of each attribute value to the subgroup divergence. *DivExplorer* algorithm extracts frequent subsets of attribute values and estimates their divergence. It begins by accepting a dataset \mathcal{D} including the ground-truth values, the prediction results from a model, and a support threshold value. Next, it examines each data point in \mathcal{D} to be a false-positive, false-negative, or otherwise, and the results are mapped into a one-hot-encoding representation. Next, depending on the **frequent pattern mining (FPM)** algorithm of choice (using off-the-shelf techniques), for each step i in FPM, itemsets with the minimum required support are extracted. Next, the cardinality of each itemset w.r.t. to the outcome function (false-positive, false-negative, or otherwise) is calculated. If the support of the itemset (sum of the cardinalities divided by the size of \mathcal{D}) is more than the specified support threshold, then the itemset is added to the list of frequent items. Once all of the frequent itemsets are determined, the outcome rate of interest (false-positive rate, false-negative rate, Accuracy, etc.) is estimated for all frequent itemsets and the divergence of all frequent itemsets as the difference outcome rate for the itemset I and the entire dataset \mathcal{D} is computed and returned.

In another related work, Farchi et al. [44] propose **Shapley Slice Ranking Mechanism with focus on Error concentration (SSR-E)** to rank data slices by the order of being problematic. However, they assume the slices are given as an input and they use the notion of Shapley value to rank the slices. They model the slices as players in a cooperative game and capture the importance of error concentration and statistical significance of the slices by defining various characteristic foundations. SSR-E accepts a model and a dataset \mathcal{D} with n slices as input. For each slice, the algorithm computes the set of data points that are misclassified by the model. The Shapley value of each slice is calculated as the independent sum of the originality of its data points. The originality of each misclassified data point is proportional to the number of slices to which it belongs. Finally, the slices are returned in a non-increasing order w.r.t. their Shapley values.

Cabrera et al. [23] propose a system called FairVis that employs a different approach to identify underrepresented subgroups in the combinatorially large space. They perform clustering on the training dataset to find similar subgroups and then use an entropy technique to find important features that are more dominant in that subgroup. When a feature's entropy is too close to zero, it means that it is concentrated in one value, which makes the feature more dominant in that subgroup. Next, they calculate a fairness score on the clusters and present the subgroups to the user sorted by the score. Once a problematic subgroup has been identified, users can compare them with similar subgroups to discover which value differences impact performance or to form more

general subgroups with fewer features. The similarity between a pair of subgroups is calculated by summing the Jensen-Shannon divergence between all features.

Finally, in Reference [73], Lees et al. suggest exploring each subpopulation's sample complexity bounds for learning an approximately fair model with a high probability. Sample complexity provides a lower limit on the count of training samples that are necessary from the subpopulations to learn a fair model. They demonstrate that a classifier can be representative of all subgroups if adequate population samples exist and the model dimensionality is aligned with subgroup population distributions. In case the sampling bias of the subpopulations is not met, human interventions in the data collection process by correcting representation bias (for example, collecting more data for under-represented subpopulations) are recommended.

Multiple Relations. In the real world, data is more commonly stored and integrated into databases with *multiple* tables. To analyze the representation bias, a combinatorial number of attribute-value combinations from different tables needs to be explored. In this process, the data to be analyzed is obtained through complex operations, e.g., table joins and predicate combinations, in databases with multiple relations. Due to the sheer data volume, determining adequate coverage can require a prohibitively long execution time. In Reference [77], Lin, et al. focus on the threshold-defined coverage identification in the multiple table scenario. Following the definition of the data pattern and MUP in the single table scenario, the coverage of a pattern P in a database with multiple relations is defined as the number of records satisfying P in the equal join result over all the tables. The coverage analysis for multiple relations has two main challenges: (1) For a given data pattern P , to determine its coverage in the database, we need to execute a conjunctive COUNT query with table joins. It would be hard for the users to enumerate the queries for all data patterns, and the execution time for a combinatorial number of such queries is prohibitive. Query optimization for the set of conjunctive COUNT queries to determine MUPs is needed for coverage analysis. (2) In the lattice space of the pattern graph, we need to design search algorithms to identify the set of MUPs with the minimum number of COUNT executions. The authors design a highly parallel index scheme to handle joins and cross-table predicate combinations to efficiently compute the number of records for each given group. As discussed in Reference [8], the MUP identification problem is an NP-hard problem. To traverse the combinatorially large search space of the pattern graph, Reference [77] designs a priority-based search algorithm that could minimize the number of computations to assess the count for a given group. The priority-based algorithm keeps searching the nodes with higher pruning efficiency. When a node is dominated by MUPs or dominates a covered pattern, it prunes this branch based on the coverage monotonicity property. The priority of the nodes is computed by a heuristic priority scoring function:

$$priority = \omega_p \times n_p + \omega_c \times n_c,$$

where n_p and n_c are the numbers of parent nodes and child nodes for each data pattern, ω_p and ω_c are the weights for parents and children. With a higher weight for child nodes, the priority algorithm would be close to top-down BFS, while with a higher weight for parent nodes, the algorithm is more likely to traverse deep to the lower layers.

Besides, as the number of patterns does not need the exact counts for the patterns, we only need to determine whether the database contains more records than the given threshold or not. Therefore, this article also provides a sampling-based approximate algorithm for coverage identification, which allows more efficient computation with smaller data sizes.

Example 4 (Priority Search Algorithm (Lin et al. 2020)). Consider the search process in Example 2, the search of the priority-based algorithm will start from the root pattern: (race = X \wedge gender = X), where X represents unspecified values. Suppose it is covered and we need to explore its children

to find the set of MUPs. Next, we evaluate all its children, suppose among its children, the pattern ($\text{race} = X \wedge \text{gender} = \text{female}$) has more descendants than the pattern ($\text{race} = \text{Asian} \wedge \text{gender} = X$) (because of the different cardinalities of race and gender.). The priority-based algorithm will first compute the coverage of the pattern ($\text{race} = X \wedge \text{gender} = \text{female}$), as once we determine its coverage, we can prune more patterns in the search process.

4.2.2 Continuous Attribute Space. Data in the real world often consists of a combination of continuous and discrete values. To better understand representation bias in continuous datasets, let us look further into our running example:

Example 5 (Representation Bias in Continuous Attribute Space). Consider a model trained on dataset \mathcal{D} described in Section 4.1. While the model can discriminate w.r.t. categorical attributes such as *sex* and *race*, it may also discriminate based on continuous-valued attributes such as *age* (e.g., because most tech workers and job applicants are young). If there are not enough entries for different *age* ranges (e.g., $\text{age} > 40$) in a dataset, it may not be trained with enough data to make a decision for those ranges.

Regarding the example above, simple solutions like binning age into “young” and “old” can transform the continuous space into discrete. However, they may lead to coarse groupings that are sensitive to the thresholds chosen. It may be inappropriate to treat a 35-year-old as young but a 36-year-old as old.

Techniques in this category assume data with continuous-valued attributes and propose solutions for identifying representation bias in such datasets.

Following a similar definition of coverage discussed in Reference [8], Asudeh et al. [9] extend the notion of coverage to continuous space for identifying representation bias. The problem of identifying representation bias using the continuous notion of coverage is defined as follows: Given dataset \mathcal{D} with n tuples over d attributes, and vicinity radius ρ and coverage threshold k , identify the uncovered region. A query point in continuous data space is covered if there are enough (at least k) data points in its ρ -vicinity neighborhood. ρ -vicinity neighborhood is the circle centered at the query point with radius ρ . The uncovered region is demarcated by the collection of all the uncovered query points in the space.

Depending on the number of attributes in a dataset, they propose two algorithms for identifying uncovered regions in data. First algorithm known as *Uncovered-2D* studies coverage over two-dimensional datasets where $X = \{x_1, x_2\}$. To find the number of circles that a query point falls into and consequently discover the uncovered region, *Uncovered-2D* makes a connection to k th order Voronoi diagrams. Consider a dataset \mathcal{D} and its corresponding k th order Voronoi diagram. For every tuple $t \in \mathcal{D}$, let \circ_t be the d -dimensional sphere (d -sphere) with radius ρ centered at t . Consider a k -voronoi cell $\mathcal{V}(S)$ in the k th order Voronoi diagram $V_k(\mathcal{D})$. Any point q inside the intersections of the d -spheres of tuples in S , i.e., $q \in \bigcap_{t \in S} \circ_t$, is covered, while all other points in the region are uncovered. The algorithm starts by constructing the k th order Voronoi diagram of the dataset and then for each Voronoi cell $\mathcal{V}(S)$ in the diagram, it computes the intersection of the circles of the tuples in S and marks the portion of $\mathcal{V}(S)$ that falls outside it as uncovered. After identifying the uncovered region, a 2D map of $\{x_1, x_2\}$ value combinations is used to report the region to the user. Let us look into how *Uncovered-2D* performs on our running example:

Example 6 (Uncovered-2D (Asudeh et al. 2021)). Consider dataset \mathcal{D} described in Section 4.1. Suppose that we are interested in identifying the uncovered region w.r.t. to hours-per-week and years-experience attributes. Suppose that a query point is covered if it has two data points in its 0.1 radii. As illustrated in Figure 4, the algorithm generates the second-order Voronoi diagram for \mathcal{D} ,

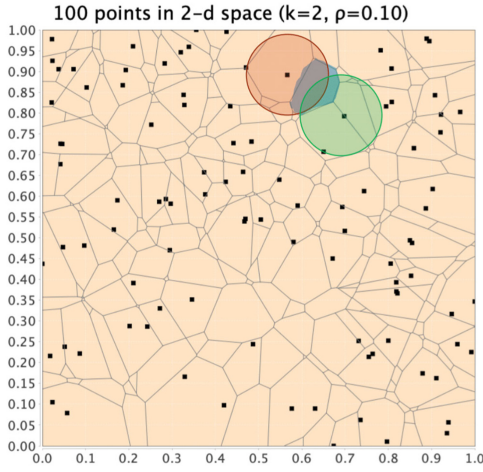


Fig. 4. Identifying the covered region in the gray Voronoi cell.

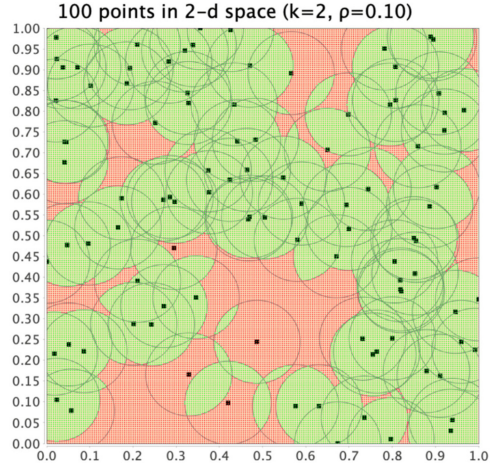


Fig. 5. Covered region in dataset \mathcal{D} marked in green. The covered region is the union of all the covered points in each Voronoi cell.

and for each Voronoi cell, the intersection of the two closest circles with radius 0.1 is considered to be the covered region. Figure 5 shows the covered and uncovered regions in \mathcal{D} .

The algorithm for the 2D case can be extended to the general case by relaxing the assumption on the number of attributes to discover the exact uncovered region, however, due to the curse of dimensionality, the search size space explodes as the number of dimensions increases and as a result, the algorithm will not be practical. Therefore, they propose a randomized approximation algorithm based on the geometric notion of ϵ -net [57]. In short, ϵ -net approximates a set using a collection of simpler subsets. Let X be a set and \mathcal{R} be a set of subsets of X . A set $N \subset X$ is an ϵ -net for X if for any range $r \in \mathcal{R}$, if $|r \cap X| > \epsilon|X|$, then r contains at least one point of N . The idea is to take random samples from the space (every sample is a potential query point) and check whether each point is covered or not and label them as +1 if uncovered and -1 otherwise. If we have enough samples in this collection, then an ϵ -net is formed, using which the uncovered region can be learned. The problem with *Uncovered-MD* is that, theoretically speaking, in adversarial cases, the number of samples may be exponentially large to the number of dimensions. However, in practice, the adversarial case is unlikely to happen, since the boundary complexity depends on the number of arcs constructing it, which can be significantly less than the theoretical upper bound provided for the number of samples.

4.3 Resolving Representation Bias

After identifying representation bias in data, the next step is presenting a remedy for it. The first approach to tackling this problem is adding more data while hoping to address the under-representation issues. However, with limited control over the data collection processes, it could be difficult and expensive for the data scientist to collect more data from the data sources. When adding more data is not feasible, the current research suggests preventive solutions such as informing the user about the representation bias issue or rewriting queries to meet the representation constraints. With that being said, we would like to emphasize the necessity of human-in-the-loop in the resolution process. It is vital to notice that not all the under-represented regions in the data are meaningful, and some may even be invalid. Therefore, a domain expert must evaluate and semantically validate the identified groups/regions.

Generally speaking, resolution techniques that operate by adding more samples to the dataset (e.g., data collection, data integration) require additional sources of data available that can be employed to resolve representation bias. Such techniques are effective when representation bias is due to the reasons such as sampling and selection bias. However, when reasons such as historical bias cause representation bias, it is costly, if not unlikely, to find additional sources to collect enough data from minorities. In such cases, preventive techniques (e.g., generating warning signals, nutritional labels, query rewriting) are effective to help the users make informed decisions.

In the following, we will introduce state-of-the-art techniques for resolving representation bias in structured data.

4.3.1 Adding More Data. Enriching the dataset with more data is the best way to address the under-representation issues. However, adding more data is not free. In particular, when the representation bias is due to the underlying distribution skew (see Section 3.1), collecting more data from the under-represented groups may violate the i.i.d. sample requirement, as the data may no longer follow the underlying distribution. Furthermore, there are not always opportunities for adding more data through data collection or integration. In these cases, the existing research has acquired techniques like data augmentation to potentially improve whatever data is available and address the lack of representation issues.

Data Collection. Data collection is usually costly. If the data are obtained from some third party, then there may be a direct monetary payment. If the data are directly collected, then there may be a data collection cost. In all cases, there is a cost to cleaning, storing, and indexing the data. To minimize these costs, as little additional data as possible should be acquired to meet the representation constraints.

In this regard, Asudeh et al. [8] suggest identifying the smallest number of additional data points needed to hit all the large uncovered spaces. Given the combinatorial number of patterns, it is not feasible to cover all of the patterns in practice. To do so, they determine the patterns for the minimum number of items that must be added to the dataset to reach a desired maximum covered level or to cover all patterns with at least a specified minimum value count. This problem translates to a hitting set instance that can be viewed as a bipartite graph with the value combinations on the left side and the uncovered patterns on the right. There is an edge between a combination and a pattern if the combination matches the pattern. The objective is to select the minimum number of nodes on the left side that hit all the patterns on the right. The hitting set problem is NP-complete, and the greedy approach to select the value combination that hits the maximum number of un-hit patterns guarantees a logarithmic approximation ratio for it.

Example 7 (Coverage Enhancement (Asudeh et al. 2019)). Consider the example in Section 4.1, consider attributes {race, marital-status, gender}, suppose the set of MUPs contains two patterns: P_1 : (race = X \wedge marital-status = Single \wedge gender = female) and P_2 : (race = Asian \wedge marital-status = X \wedge gender = female). A run of the greedy algorithm picks a pattern (race = Asian \wedge marital-status = Single \wedge gender = female), and this pattern hits both P_1 and P_2 in MUPs, therefore, the coverage enhancement process finishes.

Azzalini et al. [11] propose an approach to mitigate the representation bias in the data by adding tuples to the CFDs identified using the techniques in References [10, 12]. There are two ways to add tuples with regard to a CFD. The first option is by adding tuples to the opposite target variable of the identified CFD. As an example, if (*gender* = female, *marital-status* = single) \rightarrow *Income* = " $\leq 50k$ " is the identified CFD, then tuples should be added to (*gender* = female, *marital-status* = single) \rightarrow *income* = ">50k." The second alternative is adding to advantaged group (*gender* = male, *marital-status* = single) \rightarrow *income* = " $\leq 50k$," however, this method could cause potential issues

such as increased discrimination. The proposed algorithm to optimally add tuples to the dataset is an improved version of *Greedy Hit-Count* algorithm [8]. For each CFD with the opposite target variable, a vector of size d (d being the number of attributes in the dataset) is created, and the values of the vector are filled according to the values in the corresponding CFD or X if unspecified. Next, *Greedy Hit-Count* algorithm accepts the discovered patterns as inputs and returns the minimum set of tuples required to repair the dataset. After this step, some of the identified CFDs may still be present that can cause bias, so, in the final step, a correction algorithm removes the tuples associated with the remaining CFDs from the dataset.

In another work, Tae et al. [116] focus on acquiring the right amount of data for data slices such that both accuracy and fairness are improved. Acquiring the same amount of data for all slices may not have the same cost-benefit and it can bias the data and affect the model's accuracy for other regions. Therefore, they propose a few data acquisition strategies (including three baselines) such that the models are accurate and fair for different slices. Baselines include acquiring the same amount of data for all slices, acquiring data for all slices such that in the end they all have the same amount of data (Water-filling algorithm) and acquiring data in proportion to the original data distribution. None of the baselines solve the problem in an optimal way and in many cases increase the loss and unfairness of the models. This leads to the selective data acquisition problem that is defined as given a dataset, a set of data slices, a model trained on the dataset, a cost function for data acquisition, and a data acquisition budget, acquire examples for each slice such that the model's average loss and average unfairness over all slices are minimized while the overall cost for data collection fits the budget. The idea is to estimate the learning curves of slices, which reveal the cost benefits of data acquisition. The impact of data acquisition on the model's loss is significant at first but then gradually stabilizes to the point where it is not worth the effort anymore. Given the learning curves, *Slice Tuner* uses the learning curves to determine how much data to acquire per slice to optimize the model accuracy and fairness across the slices while using a limited data acquisition budget. However, in reality, learning curves are not perfectly generated, because slices may not have sufficient data for the model loss to be measured. Besides, acquiring data for one slice may affect the loss of the model on some other slices and eventually change their learning curves. So, it is important to generate learning curves that are reliable enough to still benefit *Slice Tuner* given these issues. The selective data acquisition problem can be considered in two different settings: For the cases where slices are independent of each other, it is only needed to solve the optimization problem once. Since the objective for minimizing loss and unfairness is global, optimization should be done on all slices. The *One-shot* algorithm updates the learning curves and solves the optimization problem to determine the amount of data that needs to be acquired for each slice. When slices are dependent, *Slice Tuner* iteratively updates the learning curves as more data is acquired. Besides, the iterative updates make the learning curves more reliable, as they are updated whenever enough influence happens, irrespective of its direction. The *Iterative algorithm* limits the change of imbalance ratio to determine the amount of data to obtain for each slice. Next, let us look into an example of how the Iterative algorithm works:

Example 8 (Iterative Algorithm (Tae et al. 2021)). Recall dataset \mathcal{D} from Section 4.1. Using the aforementioned techniques from Reference [32] slices S_1 of initial size 5 and S_2 of initial size 10 have been identified as the problematic slices. Suppose that the minimum slice size is required to be $L = 10$ and the data acquisition budget is $B = 55$. First, the iterative algorithm acquires 5 tuples for S_1 to meet the required slice size criteria, which brings down the budget B to 50 and updates the slice sizes for S_1 and S_2 to $[10, 10]$. Next, the imbalance ratio is calculated as $\frac{10}{10} = 1$. While there is still some budget left, suppose OneShot determines $[10, 40]$ tuples to be acquired for S_1 and S_2 . If all of this data is acquired, then the imbalance ratio will become $\frac{10+40}{10+10} = 2.5$. Therefore, the difference

between the imbalance ratio before and after data acquisition is $2.5 - 1 = 1.5$, which exceeds $T = 1$ (for simplicity, T is a given constant). To avoid exceeding T , change ratio x is calculated such that $\frac{10+40x}{10+10x} = 2$. With $x = 0.5$, the number of tuples to be acquired becomes $0.5 \times [10, 40] = [5, 20]$. Next, the data is acquired and budget B , and the rest of the corresponding variables are updated, and so long as there is still budget left, another iteration of OneShot and the subsequent steps are executed.

Data Augmentation. Data augmentation techniques increase the size of data by adding partially altered duplicates of already existing tuples or generating new synthetic entries from existing data. Some of the existing works adopt these techniques by adding synthetic points with different values for the attribute of interest for representation. Consequently, the new dataset has an equal number of elements for different values of the attribute of interest, resulting in potentially resolving the under-representation issues.

In Reference [107], Sharma et al. propose a novel data augmentation method to address the lack of representation of subgroups in a dataset. For a dataset with a protected attribute having a privileged and unprivileged subpopulation, they create an ideal world dataset: For every data sample, a new sample is created that has the same label and features as the original sample except that it has the opposite value for the sensitive attribute compared to the original sample (e.g., if the original sample has the sensitive attribute *gender*=male, then the new sample is *gender*=female and identical to the original sample w.r.t. the remaining attributes). The synthetic tuples are then sorted in order of their closeness to the original training distribution and added to the real dataset to create intermediate datasets. As a result, this new dataset has an equal number of entries for privileged and unprivileged sub-populations, while the label is not dependent on the protected attribute anymore, therefore potentially removing representation bias from the model built on the dataset. Although there are concerns about polluting the dataset with too many synthetic entries, by selectively adding the synthetic points that are closest to the original distribution in every increment, the user can see the effect of an augmentation technique that improves fairness while keeping the overall accuracy nearly constant.

Sometimes, the real-world training data could predominately be composed of majority examples with a small percentage of outliers or interesting minorities. For example, in applications such as fraud detection, disease diagnoses, and the detection of oil spills, the majority of the records are negative, while there is a small number of positive “interesting” records. Machine learning models trained on such imbalanced datasets are highly likely to have poor performance. Oversampling is one of the most commonly used methods to enhance the model performance in this case. The naive uniform oversampling algorithms simply duplicate the minorities uniformly at random and are subject to a higher risk of model over-fitting. The **Synthetic Minority Oversampling Technique (SMOTE)** [27] is a better alternative, which generates synthetic records of minorities based on their k -Nearest minority neighbors. There is a rich line of works that extend the SMOTE algorithm, for example, the SMOTE-borderline algorithms [54], which classified the minorities into noise, danger, and safe and only use the danger minorities for data augmentation; and the extension of SMOTE for high-dimensional data [17].

Similarly, Iosifidis et al. [59] suggest two techniques for resolving representation bias including an oversampling baseline by duplicating the instances from the minority subgroups to achieve balance. The idea of their main approach is to use SMOTE as an augmentation technique. They propose two approaches to creating the instances; first, producing instances based on a given attribute and populating the minority subgroup for a given attribute. Second, by generating instances based on a given attribute w.r.t. class, meaning that instances from the under-represented subgroup of a given attribute are generated to deal with the subgroup’s class imbalance.

Finally, Celis et al. [26] present a data preprocessing method for mitigating representation bias. The goal of this approach is to learn a distribution that resolves representation bias while remaining as close as possible to the original distribution. Learning a distribution in polynomial time to the dimension of the domain (versus domain size that can be exponential) guarantees the scalability of their method. They propose a framework based on the maximum entropy principle claiming that of all the distributions satisfying observed constraints, the distribution should be chosen that is “maximally non-committal” with regard to the current state of knowledge, meaning that it makes the fewest assumptions about the true distribution of the data. Using this principle, probabilistic models of data are learned from samples by obtaining the distribution over the domain that minimizes the KL-divergence with regards to a “prior” distribution such that its expectation follows the empirical average derived from the samples. Their approach for preprocessing data benefits from the maximum entropy framework by combining re-weighting and optimization approaches. Maximum entropy frameworks can be specified by a prior distribution and a marginal vector, providing a simple way to enforce constraints for sufficient representation. Using a re-weighting algorithm, Celis et al. specify the prior distribution by carefully choosing weights for each tuple such that desired fairness measures are satisfied and data is debiased from representation bias. Let us explain the re-weighting algorithm with an example:

Example 9 (Re-weighting (Celis et al. 2020)). Consider dataset \mathcal{D} from Section 4.1. Suppose that 3 tuples in \mathcal{D} make greater than 50k per year (class positive) and 4 tuples belonging to class make less than the amount (class negative). In the positive class, the gender of 2 of the tuples are male and 1 is female. In the negative class, the gender of 2 of the tuples are male and 2 are female. The weight of each tuple t is calculated as the number of tuples belonging to the class of t divided by the number of tuples belonging to the same class with identical gender as t . Therefore, the assigned weights for the tuples in \mathcal{D} are calculated as follows:

$$\begin{aligned} t = \text{female-positive} &\rightarrow w(t) = \frac{c(\text{positive})}{c(\text{positive}, \text{female})} = \frac{3}{1} = 3, t = \text{male-positive} \rightarrow w(t) = \frac{c(\text{positive})}{c(\text{positive}, \text{male})} = \frac{3}{2} = 1.5 \\ t = \text{female-negative} &\rightarrow w(t) = \frac{c(\text{negative})}{c(\text{negative}, \text{female})} = \frac{4}{2} = 2, t = \text{male-negative} \rightarrow w(t) = \frac{c(\text{negative})}{c(\text{negative}, \text{male})} = \frac{4}{2} = 2. \end{aligned}$$

Next, a marginal vector is chosen as the weighted average vector of samples to meet the representation rate constraints. Having defined the optimization program, they solve the dual form using the Ellipsoid algorithm, as it can be done in polynomial time in the dimension of data.

Data Integration. In data integration, data is consolidated from different sources into a single, unified view. Thus, it is a very effective solution to acquire data from different distributions such that sufficient representation is ensured for the underlying populations. However, there are sampling policy and cost-efficiency concerns that need to be examined.

In this regard, Nargesian et al. [88, 89] suggest **Data Distribution Tailoring (DT)** as resolving insufficient representation of subgroups in a dataset by integrating data from multiple sources in the most cost-effective manner such that subgroups in the dataset meet the count distribution specified by the user. Depending on our knowledge about data source distributions, *DT* can be defined from two different perspectives, first, when the user is aware of the data source sizes and the total number of tuples belonging to each subgroup, and second, when such knowledge about the data sources do not exist. For the cases when the group distributions are known, the process of collecting the target dataset is a sequence of iterative steps, where at every step, the algorithm chooses a data source, queries it, and if the obtained tuple contributes to one of the groups for which the count requirement is not yet fulfilled, then it is kept, otherwise discarded. To do so,

they first propose a **Dynamic Programming (DP)** algorithm. An optimal source at each iteration minimizes the sum of its sampling cost plus the expected cost of collecting the remaining required groups based on its sampling outcome. The dynamic programming analysis evaluates this cost recursively by considering all future sampling outcomes and selecting the optimal source in each iteration accordingly.

Example 10 (Dynamic Programming Algorithm (Nargesian et al. 2021)). Consider the dataset schema from Section 4.1. Suppose, to enrich the dataset, one would like to collect more samples from external sources \mathcal{D}_1 and \mathcal{D}_2 . \mathcal{D}_1 has 20% female and 80% male, and the sampling cost of 2. \mathcal{D}_2 has 40% female and 60% male, and the sampling cost of 3. For simplicity, suppose that we want to collect one tuple for each demographic group. The DP algorithm calculates the optimal cost $F(\text{female} = 1, \text{male} = 1)$ and decides the optimal source to query, as follows:

$$F(0, 0) = 0$$

$$F(1, 0) = \min\left(\frac{2}{0.2}, \frac{3}{0.4}\right) = 7.5 \Rightarrow \text{query } \mathcal{D}_2,$$

$$F(0, 1) = \min\left(\frac{2}{0.8}, \frac{3}{0.6}\right) = 2.5 \Rightarrow \text{query } \mathcal{D}_1,$$

$$F(1, 1) = \min(2 + 0.2F(0, 1) + 0.8F(1, 0), 3 + 0.4F(0, 1) + 0.6F(1, 0)) = 8.5 \Rightarrow \text{query } \mathcal{D}_1.$$

The drawback to the DP algorithm is that it quickly becomes intractable for cases where the minimum count requirements for the groups are not small. However, they provide a special case for when the (sensitive) attribute of interest is binary like *gender (male, female)* and the cost to query data is similar from all sources. The authors prove that the optimal selection for this special case is to query the data source with *maximum probability of obtaining a sample from the minority group*. Similar to the previous algorithm, the process of collecting the target data is a sequence of iterations where, at every iteration, we should select a data source to query. At each iteration, the algorithm finds corresponding data sources for each group and then, depending on which group is in the minority, it queries the proper data source. The algorithm stops when the count requirements of both groups are satisfied and then returns the target dataset. Finally, as an alternative to the DP algorithm, they propose an approximation algorithm for the general case. They model the problem as m instances of the “coupon collector’s problem,” where every j th instance aims to collect samples from the j th group and then, using the union bound, they come up with an upper-bound on the expected cost of this algorithm. The algorithm first identifies the minority groups and then queries its corresponding data source and updates the target data accordingly. Let us look into a simple example:

Example 11 (Coupon Collector’s (Nargesian et al. 2021)). Consider a case that we desire to collect 100 tuples for group \mathcal{G}_1 from the most cost-effective data source for \mathcal{G}_1 a.k.a. data source \mathcal{D} that has the largest $\frac{N_1}{N \cdot C}$ (N_1 is the number of tuples belonging to \mathcal{G}_1 , N is the entire number of tuples in \mathcal{D} and C is the sampling cost). Suppose that $N = 1,000$, $C = 1$ and $N_1 = 200$, therefore, the cost to collect $Q_1 = 100$ samples from \mathcal{G}_1 is bounded by $N \cdot C \cdot \ln \frac{N_1}{N_1 - Q_1} \approx 693$.

For the cases where the group distributions are unknown, Nargesian et al. model DT as a multi-armed bandit problem. Every data source is an arm and we want to select arms to collect the required tuples for each group. Every arm has an unknown distribution of different groups and a query to an arm has a cost. As the bandit strategy, they adopt **Upper Confidence Bound (UCB)** to balance exploration and exploitation. At every iteration, for every arm, UCB computes confidence intervals for the expected reward and selects the arm with the maximum upper bound of reward to be explored next. Finally, they argue that the reward of obtaining a tuple from a group

is proportional to how rare this group is across different data sources, or in other words, what the expected cost one needs to pay is to collect a tuple from that group.

Abernethy et al. [3, 4] propose an adaptive sampling algorithm that adequately represents sensitive demographic groups compared to the remaining groups. In each round, the algorithm either samples from the entire population or the population that is under-represented thus far. The decision to sample from which population depends on a sampling probability value p , which decides whether to minimize the performance loss of the model trained on the current data ($p = 1$) or minimize the fairness loss w.r.t. the under-represented group ($p = 0$). With that being said, algorithm samples with a probability of $1 - p$ from the under-represented group and with a probability of p from the entire population. Next, the sampled point will be added to the training data, and the algorithm proceeds to the next round.

Shekhar et al. [108] propose a similar adaptive sampling approach to Reference [4] based on the optimism principle to actively create a dataset that converges to min-max fair solutions. The optimism principle is used in the multi-armed bandit literature and tries to identify the hardest group to choose from. Given a fixed amount of budget, the algorithm dedicates more from the budget to the hardest groups (disadvantaged groups performing worst) w.r.t. a sensitive attribute and samples more from their distribution.

While References [4, 108] assume that collecting a fair dataset from existing sources is always attainable, this assumption may not always hold. In this regard, Niss et al. [92] propose an approach to check the feasibility of collecting a dataset from a set of available sources such that the minority groups are properly represented. To do so, the adaptive sampling is reduced to the convex hull feasibility problem, which is to determine whether a point falls in the convex hull of the means from a set of unknown distributions. Given a known variable x and a confidence value $\epsilon > 0$ and open set $x_\epsilon = \{y : \|y - x\| < \epsilon\}$, a sampling policy is feasible if there exists a $y \in x_\epsilon$ that lies in the convex hull of the means and otherwise infeasible. They study the convex hull feasibility problem in Bernoulli and Multinomial settings and devise four sampling algorithms as follows: The *uniform* algorithm at each iteration chooses from the distribution with the least samples resulting in a uniform sample size for all distributions. *LUCB Mean* chooses from the distribution with the confidence boundary farthest from x in the direction of greatest uncertainty. The direction of greatest uncertainty is the direction away from x , a distribution mean is least likely to lie on. *LUCB Ratio* chooses from the distribution whose confidence region has the biggest fraction of area on the side of x in the direction of greatest uncertainty. *Thompson Sampling*, commonly used in the multi-armed bandit literature, samples a mean from the posterior of each distribution and chooses the distribution with the mean furthest from x in the direction of greatest uncertainty.

4.3.2 No More Data Available to Add. It is not always possible to add more data to the datasets, as there might be complications such as unknown underlying distribution, lack of additional data, and so on. Existing work suggests alternative solutions to tackle these scenarios, such as informing the users about the deficiencies in the dataset or raising warnings at query time. Furthermore, by adding proper constraints on the queries w.r.t. the attributes of interest, an effort is made to ensure the proper representation.

Generating Proper Warning Signal. Generating proper signals for the trustworthiness of the analysis [9] occurs when querying about a particular data point that might potentially be concerning due to belonging to an under-represented subpopulation. The warning signal states whether the query point is covered or not. For the 2D case, the idea is to find the Voronoi cell that the query point belongs to and check the point's distance to all the points from the dataset that fall into that cell. If either of the distances is larger than the vicinity threshold, then the query point is uncovered and a warning signal is generated. For the MD case, the classifier trained on the last

iteration of the *Uncovered-MD* algorithm is used to determine the coverage of the query point by the dataset. Finally, whether to consider the outcome and how to take action is a decision left to the model user.

Data Labels and Data Sheets. Annotating datasets with representation information informs the data scientist about the potential deficiencies due to representation bias when the model is being constructed. This is a signal to investigate the fitness of data for a particular task before building the models.

In Reference [49], Gebru et al. propose a list of questions that dataset collectors should have in mind before the procedure and respond to after the collection is done. Users can then make informed decisions about the fitness of the dataset for their tasks. A number of these questions address the representativeness of the dataset such as whether the dataset includes all possible instances or is a sample (not necessarily random) of a larger set and if it is the latter, what is the larger set? Is the sample representative of the larger set, and if so how the representativeness was verified, otherwise, why not? Does the dataset identify any subpopulations such as race, gender, age group, and so on, and, if so, how are these subpopulations identified, and what is their distribution like in the dataset? Does the dataset include attributes that can be considered sensitive such as racial or ethnic origins, sexual orientations, religious beliefs, political opinions, and so on?

Some research proposes using data labels to help data users choose the appropriate datasets for their tasks. Information about data coverage is important to the dataset profiling. MithraLabel [113] provides a set of visual widgets delivering information about the dataset among different tasks on the representativeness of minorities, bias, correctness, coverage in terms of MUPs, outliers, and much more.

In References [85, 86], Moskovitch et al. design a “coverage label” of compact size that can be used to efficiently estimate the counts for each combination of discrete attributes (pattern). They provide a tradeoff between the label size and the estimation error of pattern counts. The label model is built upon an estimation function that allows the users to estimate the count of every pattern. The authors design a label for a given subset S that stores the pattern count for each possible pattern over S and the value count of each value appearing in the dataset. The identification of the optimal labels is an NP-hard problem. The authors also present an optimized heuristic for optimal label generation.

Query Rewriting. Consider a dataset with some interesting attributes (for example, gender, race, age) that are prone to be under-represented and a query over the data. Now, suppose that some representation constraints are given w.r.t. the result of a query when executed over the dataset (for example, the number of females to be greater than a given threshold), but when the query is executed over the dataset, results do not satisfy the required constraints. The idea of query rewriting is to minimally rewrite the transformation queries so certain representation constraints are guaranteed to be satisfied in the result of the transformation.

Accinelli et al. [7] propose an approach for rewriting filter and merge operations in preprocessing pipelines into the closest operation so the unprivileged groups are sufficiently represented. This is motivated by the fact that the under-representation of a subpopulation in an initial or intermediate dataset in preprocessing pipelines may lead to the under-representation of that subpopulation in any future analyses. To do so, they provide an approach that minimally rewrites the transformation operation such that coverage constraints are ensured to be met in the transformed outcome. Many potential rewritings could exist, however, their proposed sample-based approximate approach finds minimal rewriting of the original query. Queries are transformed into a canonical form as a preprocessing step. Next, the search space of potential rewritings is discretized, in such an order that an approximation of the optimal solution can be determined in

the next step, by inspecting the succeeding finite set of points. The modified input query meeting coverage constraints can be acquired by examining the grid resulting from the preprocessing step, in an order that ensures the fast identification of the closest rewriting, and by confirming constraint satisfaction using a sample-based approach. The coverage-based rewriting is approximate as a result of the discretization of the search space and of the error in estimating cardinalities and constraint satisfaction on the sample. They propose three algorithms, including a baseline for coverage-based query rewriting. **Coverage-based Rewriting Baseline (CRBase)** visits the grid in increasing order of distance from the first cell of the grid. During the visit, we look for the cell corresponding to the query with the minimum cardinality that satisfies coverage-based constraints. **CRBase with Pruning (CRBaseP)** adds some pruning rules to reduce the search space, and **CRBase with Pruning with Iteration (CRBasePI)** further optimizes *CRBaseP* by iteratively increasing the number of bins during the search up to a given maximum. As a result, each iteration increases the precision by which they refine the query and compute the cardinalities. Finally, let us demonstrate how CRBase algorithm operates on our running example:

Example 12 (CRBase (Accinelli et al. 2020)). Recall dataset \mathcal{D} from Section 4.1. Consider a simple classification task of whether or not an employee makes greater than 50k a year on individuals working more than 40 hours a week and having more than 5 years of experience. Suppose that the selection conditions $\text{hours-per-week} > 40$ and $\text{years-experience} > 5$ lead to an imbalance in the resulting dataset with 130 single and 13 married individuals while at least 70 of each group is needed. Therefore, the query needs to be rewritten such that sufficient coverage for the married group is met. The algorithm initially transforms the selection conditions into canonical form $\text{-hours-per-week} < -40$ and $\text{-years-experience} < -5$. The search space of interest is now $\text{-hours-per-week} > -40$ and $\text{-years-experience} > -5$, and the goal is to find the closest point to $Q(-5, -40)$ such that the cardinality of married individuals is greater than 70. To do so, the algorithm performs an equi-depth (or equi-width) binning with 4 bins on each dimension in the search space. In the resulting grid, each of the grid points represents an SPJ sensitive query obtained from Q by replacing selection constants with the grid point coordinates. Starting from Q , the grid points are traversed with various strategies until a point at the minimum distance from Q that meets the coverage condition is found. Suppose that this point is $Q'(-4.5, -36)$, therefore, the query is re-written as $\text{hours-per-week} > 36$ and $\text{years-experience} > 4.5$.

Since the proposed methods in Reference [7] are approximate, Accinelli et al. further expand their approach in Reference [5] by introducing some measures for computing the appearing errors. These errors include approximation error resulting from the usage of the grid for the discretization of the query search space, the approximation error correlated with the usage of a sample during the preprocessing and processing phases, and finally, the error related to the detected optimal rewriting.

As a continuum to the previous works in References [5, 7], Accinelli et al. [6] further extend the considered queries and constraints and also the proposed accuracy measures.

Shetiya et al. [109] propose a fairness-aware query rewriting approach in range queries. They use representation ratio as their measure of fairness to address selection bias and try to rewrite the original query such that the most similar results to the original query are returned while meeting the fairness criteria. Depending on the number of predicates in the query, they propose three algorithms. First, **Single Predicate Query Answering (SPQA)** algorithm for single predicate range queries benefits from index jump pointers and quickly looks up fair ranges that have a similarity of more than a threshold. Jump pointers are linear-size indices that enable sub-linear query answering time. Let us demonstrate how SPQA works using our running example:

Example 13 (SPQA (Shetiya et al. 2021)). Consider dataset \mathcal{D} from Section 4.1. Suppose that we are interested in finding individuals who work greater than 40 hours a week. By performing a selection query on the dataset, suppose that we observe a 20% difference between the number of male and female entities in the query outcome. Considering gender equity, we want to have at most a 5% difference between the number of male and female individuals. SPQA finds the most similar fair range to the input query by moving along a jump pointer. Initially, the start end-point of the range is fixed and SPQA expands the end end-point until a fair query is found. When the window indicating the start and end of the fair range is swept to the left, the start end-point can perform a shrink or an expansion. Finally, a fair query ($38 < \text{hours-per-week} < 44$) most similar to the original query is determined such that the difference in the number of male and female individuals is less than 5% while the Jaccard similarity between the two queries results is $\approx 80\%$.

Best First Search Multi-Predicate (BFSMP) algorithm models the problem of multi-predicate query answering as the traversal over a graph where nodes represent different queries and there is an edge between two nodes if their outputs differ by one tuple. Starting from the input range, BFSMP efficiently explores neighboring nodes to find the most similar fair range. Finally, inspired by the A^* algorithm, they propose **Informed BFSMP (IBFSMP)**, which improves BFSMP using an upper bound on the Jaccard similarity for effective graph exploration.

Moskovitch et al. [87] propose using the notion of provenance to mitigate bias in databases by finding minimal query relaxations that increase the number of tuples in groups satisfying a predicate. To do so, the tuples in the dataset are annotated with the query selection conditions, and annotations are propagated in the query evaluation phase. The annotated provenance value for each tuple is $\text{prov}(t) = \prod_{i=1}^{i=k} A_{i[t.A_i]}$ and the provenance inequality of the interested constraint is $Q(D)_{\mathcal{G}} = \sum_{t \in Q(D)_{\mathcal{G}}} \text{prov}(t) \geq x$. If the provenance inequality holds for a query, then the truth of the quality $\mathcal{T}_{\mathcal{P}}(p)$ is *true*. Next, using the provenance inequality, they present a method for generating minimal relaxations. They use a **minimal changes table (MCT)** with values being the terms in the provenance inequality sorted in ascending order by their minimal change w.r.t. each column. Finally, they traverse the table in a left-right top-down fashion and keep a result set that they add relaxations or remove them from.

Example 14 (Query Relaxation (Moskovitch et al. 2022)). Consider the Example in Section 4.1, suppose a query wants to select some people who are aged over 60. The fairness requirement is that the results should contain more than 5 Black female aged over 60. However, the query result only gives 3 records satisfying the condition. Query relaxation is used to relax the predicates on the continuous values in the query to include more entities in the result. A minimal relaxation is one that no other query relaxation returns a subset of it. For example, changing the search condition from Black female aged over 60 to Black female aged over 50 could be a minimal relaxation to get enough records if no other query relaxation on the age attribute is closer to the original query and satisfies the fairness requirement.

5 REPRESENTATION BIAS IN UNSTRUCTURED DATA

There has been extensive work on techniques for identifying and resolving representation bias in tabular datasets, as we have discussed above. Additionally, there is research investigating representation concerns in unstructured data types such as images, text, and graphs. In this section, we discuss the body of literature on identifying and mitigating representation bias in unstructured data.

5.1 Representation Bias in Image Data

Computer vision systems have recently achieved outstanding capacity. Identification and resolution of unwanted biases—specifically, the ones due to the disproportionate representation in the

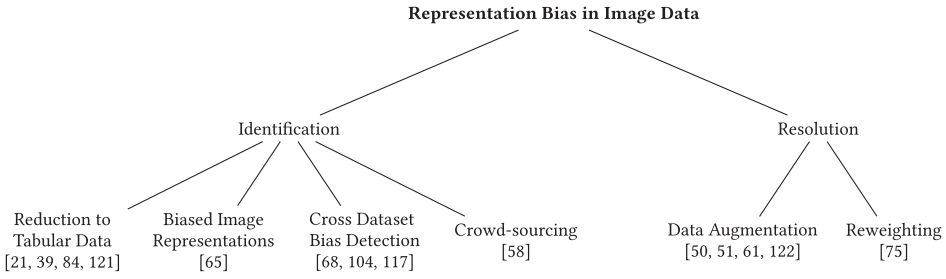


Fig. 6. Classification of techniques on identifying and resolving representation bias in image datasets.

image datasets—have drawn a lot of attention from different research communities. In this section, inspired by Fabbrizzi et al. [43], we present a taxonomy (as seen in Figure 6) to classify the techniques and, followed by its structure, we review the techniques for debiasing image datasets. Additionally, while the extent of the works studied in this section is broader than those reviewed in Reference [43], we would like to direct the interested reader to Reference [43] for a more comprehensive survey exclusively on the subject.

5.1.1 Identification of Representation Bias.

Reduction to tabular data. The main idea of this group of techniques is to transform the image data into tabular data to benefit from the rich literature on the identification of bias in tabular data. The transformation process involves direct feature extraction from the images using recognition tools and/or indirectly using the metadata of the image such as description, tag, and so on. Of course, these automatic techniques may themselves perpetuate and amplify the biases in the data, as they are prone to errors. In this regard, Dulhanty et al. [39] evaluate two subsets of ImageNet [35] with human images for representation bias w.r.t. gender and age. They first apply a face recognition algorithm to the data and, next, they apply gender and age recognition models to the outcome. With age and gender attributes determined, they calculate the distributions among genders and age groups.

Buolamwini et al. [21] created a benchmark dataset with balanced entities w.r.t. gender and skin color by counting and used it to audit the existing gender classification models.

Merler et al. [84] propose utilizing information theoretical measures of diversity and evenness such as Shannon entropy, Simpson index, and so on, to construct balanced datasets. However, they use existing recognition models or annotators for labeling the images w.r.t. gender and race.

Wang et al. [121] build a tool named REVISE for identifying and mitigating bias in visual datasets. Their scope is limited to three sets of metrics: (1) *Object-based*, which focuses on statistics about object frequency, scale, context, or diversity of representation; (2) *Person-based*, which examines the representation of people from various demographics in the dataset and allows the user to assess what potential downstream consequences this may have to consider how best to intervene. It also builds on the object-based analysis by considering how the representation of objects with people of different demographic groups differs. (3) *Geography-based*, which considers the portrayal of different geographic regions within the dataset and is deeply intertwined with the previous two, as geography influences both the types of objects that are represented, as well as the different people that are pictured. REVISE accepts annotated image datasets as input and, depending on the annotations, it provides insights on the datasets based on each of the three categories of metrics explained above: metrics such as object count, scale, co-occurrence, scene diversity, and so on, for *Object-based* category; person prominence, appearance differences, and contextual representations for *Person-based*; and geography distributions based on people, language, weather, and so on, for

Geography-based category. REVISE does not claim to find all the visual biases, and it is limited to the available annotations accompanying the data.

Biased Image Representations. The techniques in this group use distance-based analysis on the low-dimensional representation of the images in the embedding space to identify representation bias. Particularly, Karkkainen et al. [65] create a balanced face dataset w.r.t. age, race, and gender. To evaluate the diversity of their dataset compared to the existing work, they visualize the images in 2D using t-SNE [118], a statistical method for visualizing high-dimensional data by giving each data point a location in 2D/3D space on the embeddings trained on multiple online sources. Next, they measure pairwise Manhattan distances between random subsets of the images based on their 128-dimensional embedding. The skewness of the resulting distribution towards high distances is evidence of high diversity and proper representation of different subgroups.

Cross Dataset Bias Detection. Each dataset includes specific *signature* biases that make it distinct from the rest. This signature bias is introduced in the data collection process and affects the generalizability of the models built on the dataset. This group of methods evaluates the signature bias by comparing different datasets.

In this regard, Torralba et al. [117] perform some experiments on famous image datasets to measure the bias. To correctly measure the bias of a dataset, it should be compared to the real visual world, which would have to be in the form of a dataset, which could also be biased and, consequently, not a viable option. Therefore, they suggest *Cross-dataset Generalization* by training a model on a dataset and testing it on another. Assuming that the training dataset is truly representative of the real world, the model should perform well; otherwise, it means that there are biases, such as selection and capture, present in the dataset. Next, knowing that datasets define a visual phenomenon not only by what it is but also by what it is not, they argue about *Negative Set Bias* and whether the negative samples are representative of the rest of the world or even sufficient. To do so, they run an experiment such that for each dataset, a classifier is trained on its own set of positive and negative instances, and then during testing, the positives come from that dataset, but the negatives come from all datasets combined. The performance of the models shows how well the dataset is representing the rest of the world.

Khosla et al. [68] propose an algorithm that learns the *visual world* model and the biases for each dataset. The key observation is that all datasets are sampled from a common *visual world* (a more general dataset). A model trained on this dataset would have the best generalization ability, however, making such a dataset is not realistic. Therefore, they suggest defining the biases associated with each dataset and approximating the weights for the visual world by removing the bias from each dataset. The visual world model performs well on average but is not necessarily the best on any specific dataset, since it is not biased towards any one dataset. However, the biased model, built by combining the visual world model and the learned bias, performs superior on the dataset that it is biased towards but does not necessarily extend to the rest of the datasets. Hence, they propose a maxed-margin learning discriminative framework to collectively learn the weight vector correlated to the visual world object model and a set of bias vectors for each dataset, such that when combined with the visual world weights lead to an object model specific to the dataset.

Another related work by Schaaf et al. [104] focuses on measuring bias in image classification tasks by means of attribution maps. Attribution maps seek to explain image classification models, such as CNNs, by demonstrating the importance of each individual pixel of the input image on the outcome. To do so, they propose a four-step process to indicate their usefulness. First, they generate artificial datasets with a known bias. For example, they generate a biased fruit dataset where apples are all on tree backgrounds, while other fruits have different backgrounds, and an unbiased dataset where all fruits have different backgrounds. Next, they train biased CNN models

and then generate attribute maps using different attribution techniques such as *Grad-CAM*, *Score-CAM*, *Integrated Gradients*, and *epsilon-LRP*. Finally, they quantitatively evaluate attribution maps' ability to detect bias using metrics such as **Relevance Mass Accuracy (RMA)**, **Relevance Rank Accuracy (RRA)**, and **Area Over the Perturbation Curve (AOPC)**. Their results partly confirm the ability of attribution maps to quantify bias. However, in some cases, attribution maps provide inconsistent results for different metrics.

Crowd-sourcing. Hu et al. [58] propose a crowd-sourcing workflow to facilitate sampling bias discovery in visual datasets with the help of human-in-the-loop. This workflow takes a visual dataset as input and outputs a list of potential biases of the dataset. There are three steps in this workflow. The first step is *Question Generation* in which the crowd inspects random samples of images from the input dataset and describes their similarity using a question-answer pair. The next step is *Answer Collection*, in which the crowd reviews separate random samples of images from the input dataset and provides answers to questions generated in the earlier step. Finally, in the third step called *Bias Judgement*, the crowd judges if the statements about the visual dataset automatically generated through the accurate questions and answers collected in the former steps reflect the real world.

5.1.2 Resolving Representation Bias.

Data Augmentation. This group of techniques tries to mitigate bias by adding samples for the underrepresented groups benefiting from the rich literature on image augmentation.

Jaipuria et al. [61] propose a bias mitigation approach by using targeted synthetic data augmentation that combines the advantages of gaming engine simulations and *sim2real* style transfer techniques to bridge the gaps in real datasets for vision tasks. However, instead of blindly collecting more data or mixing datasets that often end up in worse final performance, they suggest a smarter approach to augment data regarding the task-specific noise factors. The results consistently indicate that through adding synthetic data to the training set, a noticeable improvement occurs in cross-dataset generalization, in contrast to merely training on original data, for a training set of equal size.

Georgopoulos et al. [50] propose a style transfer approach based on **generative adversarial networks (GANs)**, capable of creating additional images, reflecting multiple attributes such as race, gender, and age. The resulting dataset is less biased w.r.t. the aforementioned attributes. This is accomplished by relaxing the strict reliance on a single attribute label and adding a tensor-based mixing structure that multilinearly represents multiplicative interactions between attributes.

Similarly, Yucer et al. [122] propose another adversarial augmentation method utilizing CycleGANs to transfer race to mitigate representation bias. They aim to create a synthesized dataset by transforming facial images into different racial domains while maintaining identity-related traits so race-related traits eventually become irrelevant in determining the subject's identity.

Goel et al. [51] propose an advanced augmentation approach that is oblivious to the differences within subgroups and aims for class information shared by subgroups. In this regard, they propose **CycleGAN Augmented Model Patching (CAMEL)**, which first learns mappings between pairs of subgroups using CycleGANs and creates transformations that can be used to generate augmented examples based on the training instances and, second, leverages the transformations as data augmentations and builds a more robust classifier.

Reweighting. Li et al. [75] propose *REPAIR*, a resampling-based bias mitigation approach that is formulated as an optimization problem. *REPAIR* assigns a weight to the instances that the classifier built on a feature representation can penalize more easily. This is implemented through a deep neural network as a feature extractor for the representation of interest and learning an independent

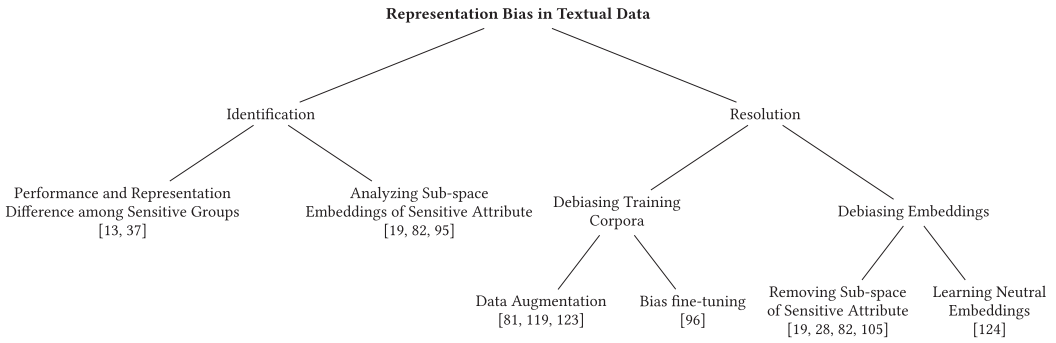


Fig. 7. Classification of techniques on identifying and resolving representation bias in textual data.

linear classifier to classify the extracted features. Next, bias mitigation is defined as maximizing the ratio between the loss of the classifier on the reweighted dataset and the uncertainty of the ground-truth labels. Last, the problem is reduced to a minimax problem, which can be solved by alternately updating the classifier coefficients and the dataset resampling weights, through stochastic gradient descent.

5.2 Representation Bias in Natural Language Data

Natural language processing (NLP) is one of the areas that has been widely affected by the data explosion and advancement of data-driven decision-making systems. However, the existing biases in the data have regularly resulted in discriminatory outcomes w.r.t. gender, race, age, disability, and so on. Representation bias as one of the key reasons for such issues has been extensively studied in different NLP tasks such as machine translation, caption generation, sentiment analysis, hate speech detection, coreference resolution, language models, and word embeddings. Hundreds of technical papers with a variety of solutions and dozens of reviews have been published tackling different angles of the matter w.r.t. the task and the target of the bias. Going through, the details of each work is out of the scope of this survey due to the richness of existing surveys [18, 36, 48, 103, 114, 120], however, we try to give an overview and a taxonomy of the techniques (as seen in Figure 7) on identifying and mitigating representation bias in textual data while giving proper directions to the curious reader.

Representation bias in textual data can happen as a result of the following [114]:

- *Denigration*: Using culturally or historically derogatory words.
- *Stereotyping*: Heightening the existing societal stereotypes.
- *Under-representation*: Disproportionately low representation of a specific group.

Each NLP task can be associated with one or more of these classes, as demonstrated in Reference [114]. Next, inspired by Reference [114], we present a taxonomy for the classification of techniques for identifying and mitigating representation bias in textual data and, following the structure of the taxonomy, we provide a summary of the techniques in the latter sections.

5.2.1 Identification of Representation Bias. There are two major approaches for identifying representation bias in the NLP literature:

Performance and Representation Difference among Sensitive Groups. Regardless of the task, most NLP model predictions should not be significantly affected by a sensitive attribute such as gender, race, and so on, of the entity. Following this fact and regarding representation bias in

the context of gender, gender swapping and measuring the difference in evaluation score (such as false-positive rate difference or false-negative rate difference) is a common practice to assess gender bias in such tasks. Furthermore, standard evaluation datasets commonly used in NLP are not sufficient for measuring gender bias, as they often contain bias themselves due to the disproportionate representation of male and female entities. Therefore, carefully designed task-specific datasets known as **Gender Bias Evaluation Test Sets (GBETs)** are constructed that can control the effect of gender bias.

Aside from the performance aspect, Dixon et al. [37] show how imbalances in the training data w.r.t. representation can lead to biases in the constructed text classification models with potentially unfair results towards the under-represented group. An example of such biases can be seen in toxicity detection models where due to disproportionate representation of terms such as “gay” in the training data, statements such as “I’m a gay man” are assigned overly high toxicity scores even though the comment is not toxic. Models are falsely biased toward words that are disproportionately represented in toxic comments compared to the overall dataset, and also they tend to be more biased toward short comments. To identify the representation bias, Dixon et al. [37] create a hand-curated list of words for which they study these two properties. Badjatiya et al. [13] add two more strategies to what Dixon et al. [37] proposed to identify representation bias in textual data. The first strategy is investigating skewed occurrences across classes. If a term happens to appear in lots of training samples belonging to the toxic class, then it encourages the models to classify a comment containing that particular term as toxic. The second strategy is skewed predicted class probability distribution, which is the maximum probability of a term belonging to a non-neutral class. A high probability value means that the model has stereotyped the term to belong to the toxic/non-toxic class.

Analyzing Sub-space Embeddings of Sensitive Attribute. Word embeddings and language models are trained on the available biased text corpora and tend to amplify and propagate these biases to the downstream tasks when used as features. Bolukbasi et al. [19] investigate representation bias in the context of gender in the embedding space by showing that, geometrically, gender bias can be captured by a direction. Besides, they show that gender-neutral words (e.g., nurse) are linearly separable from gender-defined words (e.g., queen). Therefore, it is possible to differentiate between the two and capture gender bias in the embedding space. The proposed technique operates as follows: Initially, a set of gender-specific words such as {he, she, man, woman, ...} are chosen as seed words. Using the seed words, an SVM classifier is trained to get the rest of the gender-specific words. The complement of the gender-specific corpus grants us the set of gender-neutral words. Having the gender-specific and gender-neutral words separated, they select the seed word pairs such as he-she to act as the x-axis to identify the gender subspace. By checking the distance of gender-neutral words from the he or she end of the axis (“nurse” closer to she, “genius” closer to he), they identify how biased the word embeddings are toward such words. These biases originate from the insufficient association of such words with the opposite gender in the original corpora on which the embeddings were trained.

Manzini et al. [82] extend this solution to non-binary gender and multi-class sensitive attributes such as race, religion, and so on. Papakyriakopoulos et al. [95] study detecting representation bias resulting from historical biases reflected in the word embeddings. To detect the bias of word embeddings, they define an inter-group direction (for instance, between man and woman) and then the bias is quantified as the cosine distance between the word vector and the inter-group direction. This method compares the magnitude of dependence between a concept and the two groups. If the concept vector has a higher similarity to a group than another, then the concept is considered to be biased in that direction.

5.2.2 Resolving Representation Bias. Several methods have been proposed to mitigate representation bias in textual data. Some of these methods require the models to be retrained after the alterations, while some do not and only manipulate the model to fix the outcomes. In the following, we will introduce each of these methods and reiterate some of the adopted techniques:

Text Corpus Alteration. To debias the text corpora, two approaches have been proposed:

Data Augmentation: The augmentation approach is to add modified copies of the existing data, or newly created synthetic data, to the corpora. While some works propose completely removing, masking, or replacing any indication of gender, race, and so on, from the text corpora to eliminate representation bias, De Arteaga et al. [34] make an interesting observation that even by removing the explicit indicators regarding gender, race, or socioeconomic status in the text corpora, although a slight reduction in representation bias would occur towards the minority group, a significant gap remains due to the imbalances in the available data between the minority and majority group. Similarly, Li et al. [81] make a closely related conclusion for the task of text generation where they investigate representation bias in the stories generated by GPT-3. They demonstrate how gender stereotypes occur in generated narratives, even in the absence of gender indicators or stereotype-related cues. They propose prompt design as a possible workaround for mitigating bias and steering GPT-3, however, they state that it is not a feasible solution for every situation. Zhao et al. [123] propose another approach to decrease the bias in text corpora by creating an identical but gender-swapped version of the original dataset and training the model on the union of the original dataset, the gender-swapped version and the named-entity anonymized version of the original dataset.

In tasks such as machine translation, due to the domination of male entities in the available text corpora, the models tend to predict the entities more as male while the actual gender may not be clear. This specifically becomes problematic while translating into languages such as French, where words are gender-specific and masking or removal of gender indicators is not an option. Vanmassenhove et al. [119] propose an augmentation technique known as gender-tagging that tries to solve the aforementioned issue by appending the gender of the entity to the sentences. Gender-tagging preserves the gender of the speaker and, therefore, the machine translation model can consider it while making predictions.

Bias Fine-tuning: An alternative approach to debias text corpora, proposed by Park et al. [96], is to use transfer learning from an already bias-free dataset and fine-tune on the biased data to train a model. This approach enables the models to benefit from bias-free datasets while still sufficiently good to perform the assigned learning task.

Word-embedding Adjustment. Complete elimination of representation bias from embedding space is not a feasible goal. However, it has been shown that it is possible to mitigate it w.r.t. the similarity to sensitive attribute subspace and not needing the embeddings to be retrained. To debias the word embeddings, two approaches have been proposed:

Removing Sub-space of Sensitive Attribute: This is achieved by building a neutral (i.e., genderless, raceless, etc.) framework for all words [105] or for gender-neutral words [19]. For instance, Bolukbasi et al. [19] propose a neutralization method to debias the word embeddings. Recall that to identify the bias, they projected each gender-neutral word vector on an axis with gender-specific words on each end. Having known that the bias exists, they project the gender-neutral words on the y-axis and thus eliminate the gender bias. Another approach is to make gender-neutral words equidistant to all words in the gender-specific set, meaning that the word “nurse” will be equidistant to sets {he, she} and {man, woman}. Manzini et al. [82] show that this solution is extendable to non-binary sensitive attributes.

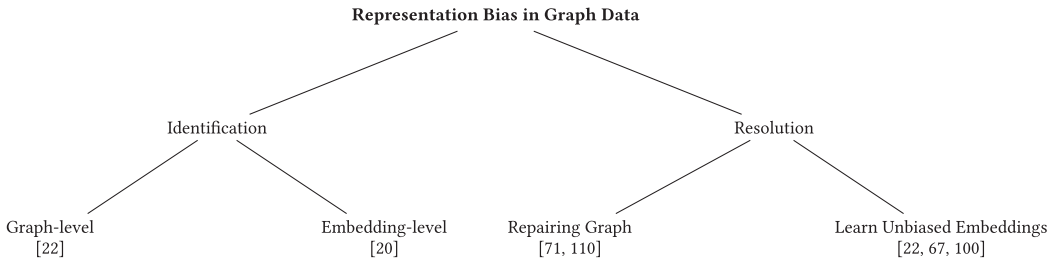


Fig. 8. Classification of techniques on identifying and resolving representation bias in graphs.

However, Cheng et al. [29] show that bias w.r.t. different sensitive attributes can be correlated and independent removal of bias may not be sufficient. To mitigate the bias at a word-embedding level, for each bias-sensitive word, they define a sentiment direction by forming pairs showing different ends of bias (e.g., good-bad, positive-negative) and taking the difference between the word embeddings of words in each set and the mean word embedding over the set. Next, they apply PCA, with the resulting component being the sentiment direction. Next, they define a corresponding set to the neutral words vector (e.g., doctor, nurse) and hard-neutralize this vector by making it orthogonal to the sentiment vector [19].

Learning Neutral Embeddings: Zhao et al. [124] suggest separating information about the sensitive attribute in a dimension and keeping the neutral information in other dimensions. In doing so, the sensitive attribute information can be utilized or neglected on demand. This method requires retraining the embeddings.

5.2.3 Representation Bias in Speech Recognition. Identification and mitigation of representation bias in speech recognition systems have been briefly studied in the contexts of gender, race, and age [46, 70, 80]. The primary approach to identifying the bias in such systems is by measuring the error rate of the speech recognition model among different subgroups. Demographic information of the speaker is usually acquired through annotations or utilizing automatic methods [30, 42, 101]. With the demographic information available, the problem is reduced to bias identification in tabular data. For the purpose of bias mitigation, diversifying the training datasets w.r.t. race, gender, age, and so on, through the addition of more data is recommended.

5.3 Representation Bias in Graphs

The capacity of graphs to model complex phenomena is gaining increasing attention in many domains, including those with high societal impact. The sensitivity of applications such as online polarization, job recommendation systems, disaster response, and criminal justice has led to increasing interest in addressing bias in these systems. There now are comprehensive studies in the form of review papers and tutorials [31, 38, 64] to identify biases and promote fairness. In this section, inspired by Choudhary et al. [31], we discuss recent techniques to identify and mitigate representation bias in graphs, present a taxonomy (as seen in Figure 8) of such techniques, and give pointers to the interested reader.

Graphs hold properties such as being non-i.i.d. and non-Euclidean that make the existing bias identification and mitigation solutions ineffective. The non-i.i.d. assumption suggests that an alteration in one node or edge will affect its neighbors in the graph. The non-Euclidean assumption states that before performing any learning task, a vectorized representation of the level of interest (node-level, edge-level, or graph-level) should be learned. Aside from the pre-existing bias in the

graphs, different objective functions to learn the representations can perpetuate and amplify the biases in the graph embeddings. The embeddings should hold two properties:

- They should reflect the properties of the graph structure.
- They should be independent of the sensitive attributes.

The first property is guaranteed through the choice of the objective function, however, the second is our problem of interest and can be secured in a two-staged process of identification and mitigation of bias in a variety of methods.

5.3.1 Identification of Representation Bias. This set of methods targets representation bias from two different levels:

Graph-level. Assortative mixing coefficient [90] is a notion that is used in Reference [22] to evaluate the homophily of a graph regarding a particular attribute. This notion is used to evaluate the graph structures for the existing biases. The values of the assortative mixing coefficient fall into a range of $[-1, 1]$, and the closer the value to -1 or 1 , the more correlated the graph is with a sensitive attribute. The mixing coefficient is calculated using the following formula:

$$r = \frac{\sum_i e_{ij} - \sum_i a_i b_i}{1 - \sum_i a_i b_i}$$

where:

$$e_{ij} = \frac{\text{card}\{(i, j) \in \mathcal{E}; A_{v_i} = i, A_{v_j} = j\}}{m}, \quad a_i = \sum_j e_{ij} \text{ and } b_j = \sum_i e_{ij},$$

where v_i is i th vertex, \mathcal{E} is the set of all edges, m is the number of all edges, A is the sensitive attribute, and a_i, b_j is the ratio of the edges starting from and ending at each of the attribute values. An r value of zero indicates no bias in the graph. Mixing coefficient value r can be calculated on any graph to determine bias and promote fairness.

Embedding-level. Representation Bias (RB) [20] (should not be mistaken with the topic of our survey though) refers to the bias in node-level embeddings. RB is calculated using the following:

$$RB = \sum_{a=0}^l \frac{1}{|V_a|} \text{AUC}(\{\mathbb{P}_h(a, z_v) | \forall v \in V_a\}),$$

where $V_a = \{v | A(v) = a\}$ is the set of nodes having sensitive attribute value a , h is a classifier trained to predict sensitive attribute A , and $\mathbb{P}_h(a, z_v)$ is the result of the classification. The idea is to consider the sensitive attribute A as the target variable and then the aforementioned formula calculates the weighted average of the one-vs.-rest AUC values from the output of the classifier trained to predict A . RB values fall into $[0, 1]$ range. The closer the value to 0.5 , the more nondiscriminatory the graph is w.r.t. the sensitive attribute.

5.3.2 Resolving Representation Bias.

Repairing Graph. The methods introduced in this section try to remove the bias from the graph structure itself rather than the embeddings. Laclau et al. [71] try to mitigate the bias in the graph structure using optimal transport technique in the context of fair edge prediction. They reduce the problem to the problem of alignment between node distributions of nodes belonging to different sensitive groups based on the rows in the normalized adjacency matrix. Accordingly, Spinelli et al. [110] propose a method to modify the adjacency matrix at the training time to balance the homophily caused by the sensitive attribute. In each training iteration, they remove the edges

between nodes based on a randomized response mechanism between nodes that have the same sensitive attribute value.

Learning Unbiased Embeddings. The high-level idea of resolving bias for the methods in this section is to place a fairness constraint on the objective function of the representation learning model. Rahman et al. [100] try to promote fairness to Node2vec [52] by modifying the random walks by changing the transition probabilities to generate unbiased traces. In consequence, the generated random walk is more likely to have nodes from different groups. Khajehnejad et al. [67] propose a re-weighting approach for generating the random walks, however, they assign more weights to the links that connect nodes from different groups to provide a higher chance of discovery in extreme cases that Rahman et al. [100] would have failed. Inspired by Conditional Network Embeddings [63], Buyl et al. [22] present a Bayesian approach that learns debiased representations using as strongly biased as possible prior so the learned embeddings have minimal information about sensitive attributes in the training step.

6 CONCLUSION

In this article, we surveyed techniques for the identification and resolution of representation bias in data. After reviewing the fairness literature at a high level, we provided a thorough overview of the problem definition, the causes, and how to measure and quantify this phenomenon in both structured and unstructured data. Depending on the data type, we then presented taxonomies based on multiple dimensions and had side-by-side comparisons of the techniques. We discussed the details of several algorithms to illustrate the different challenges and the problems they address. Two promising research directions we envision being important are:

- *Addressing representation bias in other types of datasets.* As we discussed in Section 5, with the extension of the problem scope to new data types such as streaming data, spatio-temporal data, and so on, new challenges arise and the current solutions may not be directly extendable.
- *More metrics for measuring representation bias.* Existing works have introduced *coverage* and *representation rate* for measuring representation bias. However, each metric has potential shortcomings that provide new research opportunities. Furthermore, when it comes to data quality and trust measures in data, there is no such thing as “enough” and there is always room for improvement.

REFERENCES

- [1] 2019. *Health, United States Spotlight: Race and Ethnic Disparities in Heart Disease*. Health, United States spotlight, CDC Stacks Public Health Publications. <https://stacks.cdc.gov/view/cdc/77732>.
- [2] 2019. Asian-American and Pacific Islander Heritage in the United States. <https://www.census.gov/newsroom/facts-for-features/2019/asian-american-pacific-islander.html>. Accessed 26-03-2023.
- [3] Jacob Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. 2020. Active sampling for min-max fairness. *arXiv preprint arXiv:2006.06879* (2020).
- [4] Jacob Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, and Jie Zhang. 2020. Adaptive sampling to reduce disparate performance. *arXiv e-prints* (2020), arXiv–2006.
- [5] Chiara Accinelli, Barbara Catania, Giovanna Guerrini, and Simone Minisi. 2021. The impact of rewriting on coverage constraint satisfaction. In *Proceedings of the EDBT/ICDT Workshops*.
- [6] Chiara Accinelli, Barbara Catania, Giovanna Guerrini, and Simone Minisi. 2022. A coverage-based approach to nondiscrimination-aware data transformation. *J. Data and Information Quality* 14, 4 (December 2022), 26 pages. <https://doi.org/10.1145/3546913>
- [7] Chiara Accinelli, Simone Minisi, and Barbara Catania. 2020. Coverage-based rewriting for data preparation. In *Proceedings of the EDBT/ICDT Workshops*.
- [8] Abolfazl Asudeh, Zhongjun Jin, and H. V. Jagadish. 2019. Assessing and remedying coverage for a given dataset. In *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 554–565.

- [9] Abolfazl Asudeh, Nima Shahbazi, Zhongjun Jin, and H. V. Jagadish. 2021. Identifying insufficient data coverage for ordinal continuous-valued attributes. In *Proceedings of the SIGMOD Conference*. ACM.
- [10] Fabio Azzalini, Chiara Criscuolo, and Letizia Tanca. 2021. FAIR-DB: Functional dependencies to discover data bias. In *Proceedings of the EDBT/ICDT Workshops*.
- [11] Fabio Azzalini, Chiara Criscuolo, and Letizia Tanca. 2022. E-FAIR-DB: functional dependencies to discover data bias and enhance data equity. *ACM Journal of Data and Information Quality* 14, 4 (2022), 1–26.
- [12] Fabio Azzalini, Chiara Criscuolo, and Letizia Tanca. 2022. FAIR-DB: A system to discover unfairness in datasets. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 3494–3497.
- [13] Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *Proceedings of the World Wide Web Conference*. 49–59.
- [14] Agathe Balayn, Christoph Lofi, and Geert-Jan Houben. 2021. Managing bias and unfairness in data for decision support: A survey of machine learning and data engineering approaches to identify and mitigate bias and unfairness within data management and analytics systems. *VLDB J.* 30, 5 (2021), 739–768.
- [15] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. Fairness and machine learning: Limitations and opportunities. Retrieved from fairmlbook.org.
- [16] Solon Barocas and Andrew D. Selbst. 2016. Big data’s disparate impact. *Calif. L. Rev.* 104 (2016), 671.
- [17] Rok Blagus and Lara Lusa. 2013. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinform.* 14 (2013), 106. DOI : <https://doi.org/10.1186/1471-2105-14-106>
- [18] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of “bias” in NLP. *arXiv preprint arXiv:2005.14050* (2020).
- [19] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Adv. Neural Inf. Process. Syst.* 29 (2016).
- [20] Avishek Bose and William Hamilton. 2019. Compositional fairness constraints for graph embeddings. In *Proceedings of the International Conference on Machine Learning*. PMLR, 715–724.
- [21] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the Conference on Fairness, Accountability and Transparency*. PMLR, 77–91.
- [22] Maarten Buyt and Tijl De Bie. 2020. Debayes: A Bayesian method for debiasing network embeddings. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1220–1229.
- [23] Ángel Alexander Cabrera, Will Epperson, Fred Hohman, Minsuk Kahng, Jamie Morgenstern, and Duen Horng Chau. 2019. FairVis: Visual analytics for discovering intersectional bias in machine learning. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 46–56.
- [24] Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese. 2015. Relaxed functional dependencies—A survey of approaches. *IEEE Trans. Knowl. Data Eng.* 28, 1 (2015), 147–165.
- [25] Barbara Catania, Giovanna Guerrini, and Chiara Accinelli. 2022. Fairness & friends in the data science era. *AI & Societ.* (2022), 1–11. <https://doi.org/10.1007/s00146-022-01472-5>
- [26] L. Elisa Celis, Vijay Keswani, and Nisheeth Vishnoi. 2020. Data preprocessing to mitigate bias: A maximum entropy based approach. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1349–1359.
- [27] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* 16 (2002), 321–357. DOI : <https://doi.org/10.1613/jair.953>
- [28] Irene Chen, Fredrik D. Johansson, and David Sontag. 2018. Why is my classifier discriminatory? *arXiv preprint arXiv:1805.12002* (2018).
- [29] Lu Cheng, Suyu Ge, and Huan Liu. 2022. Toward understanding bias correlations for mitigation in NLP. *arXiv preprint arXiv:2205.12391* (2022).
- [30] Donald G. Childers and Ke Wu. 1991. Gender recognition from speech. Part II: Fine analysis. *J. Acoust. Societ. Amer.* 90, 4 (1991), 1841–1856.
- [31] Manvi Choudhary, Charlotte Laclau, and Christine Largeron. 2022. A survey on fairness for machine learning on graphs. *arXiv preprint arXiv:2205.05396* (2022).
- [32] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. 2019. Slice finder: Automated data slicing for model validation. In *Proceedings of the IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1550–1553.
- [33] Sabyasachi Dash, Sushil Kumar Shakyawar, Mohit Sharma, and Sandeep Kaushik. 2019. Big data in healthcare: Management, analysis and future prospects. *J. Big Data* 6, 1 (2019), 1–25.
- [34] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnamurthy Kenthapadi, and Adam Tauman Kalai. 2019. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 120–128.

- [35] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255.
- [36] Mark Díaz, Isaac Johnson, Amanda Lazar, Anne Marie Piper, and Darren Gergle. 2018. Addressing age-related bias in sentiment analysis. In *Proceedings of the Chi Conference on Human Factors in Computing Systems*. 1–14.
- [37] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 67–73.
- [38] Yushun Dong, Jing Ma, Chen Chen, and Jundong Li. 2022. Fairness in graph mining: A survey. *arXiv preprint arXiv:2204.09888* (2022).
- [39] Chris Dulhanty and Alexander Wong. 2019. Auditing ImageNet: Towards a model-driven framework for annotating demographic attributes of large-scale image datasets. *arXiv preprint arXiv:1905.01347* (2019).
- [40] Jeff Edmonds, Jarek Gryz, Dongming Liang, and Renée J. Miller. 2003. Mining for empty spaces in large data sets. *Theor. Comput. Sci.* 296, 3 (2003), 435–452. DOI: [https://doi.org/10.1016/S0304-3975\(02\)00738-7](https://doi.org/10.1016/S0304-3975(02)00738-7)
- [41] Kathleen M. Egan, D. Trichopoulos, M. J. Stampfer, W. C. Willett, P. A. Newcomb, A. Trentham-Dietz, M. P. Longnecker, and J. A. Baron. 1996. Jewish religion and risk of breast cancer. *Lancet* 347, 9016 (1996), 1645–1646.
- [42] Hasan Erokyar. 2014. Age and gender recognition for speech applications based on support vector machines. *USF Tampa Graduate Theses and Dissertations*. <https://digitalcommons.usf.edu/etd/5356>.
- [43] Simone Fabbri, Symeon Papadopoulos, Eirini Ntoutsi, and Ioannis Kompatsiaris. 2021. A survey on bias in visual datasets. *arXiv preprint arXiv:2107.07919* (2021).
- [44] Eitan Farchi, Ramasuri Narayanam, and Lokesh Nagalapatti. 2021. Ranking data slices for ML model validation: A Shapley value approach. In *Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1937–1942.
- [45] Sara Feijo. 2020. Here’s what happened when Boston tried to assign students good schools close to home. 2020. <https://news.northeastern.edu/2018/07/16/heres-what-happened-when-boston-tried-to-assign-students-good-schools-close-to-home/>. Accessed 26-03-2023.
- [46] Siyuan Feng, Olya Kudina, Bence Mark Halpern, and Odette Scharenborg. 2021. Quantifying bias in automatic speech recognition. *arXiv preprint arXiv:2103.15122* (2021).
- [47] Donatella Firmani, Letizia Tanca, and Riccardo Torlone. 2019. Ethical dimensions for data quality. *J. Data Inf. Qual.* 12, 1 (2019), 1–5.
- [48] Tanmay Garg, Sarah Masud, Tharun Suresh, and Tanmoy Chakraborty. 2022. Handling bias in toxic speech detection: A survey. *arXiv preprint arXiv:2202.00126* (2022).
- [49] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010* (2018).
- [50] Markos Georgopoulos, James Oldfield, Mihalis A. Nicolaou, Yannis Panagakis, and Maja Pantic. 2021. Mitigating demographic bias in facial datasets with style-based multi-attribute transfer. *Int. J. Comput. Vis.* 129, 7 (2021), 2288–2307.
- [51] Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. 2020. Model patching: Closing the subgroup performance gap with data augmentation. *arXiv preprint arXiv:2008.06775* (2020).
- [52] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 855–864.
- [53] Martyn Hammersley and Roger Gomm. 1997. Bias in social research. *Sociolog. Res. Onl.* 2, 1 (1997), 7–19.
- [54] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proceedings of the International Conference on Intelligent Computing: Advances in Intelligent Computing*. Springer, 878–887.
- [55] Emma J. Harding, Elizabeth S. Paul, and Michael Mendl. 2004. Cognitive bias and affective state. *Nature* 427, 6972 (2004), 312–312.
- [56] Martie G. Haselton, Daniel Nettle, and Damian R. Murray. 2015. The Evolution of Cognitive Bias. In *The Handbook of Evolutionary Psychology*. John Wiley & Sons, Ltd, 1–20. DOI: <https://doi.org/10.1002/9781119125563.evpsych241>
- [57] David Haussler and Emo Welzl. 1986. Epsilon-nets and simplex range queries. In *Proceedings of the 2nd Annual Symposium on Computational Geometry*. 61–71.
- [58] Xiao Hu, Haobo Wang, Anirudh Vegesana, Somesh Dube, Kaiwen Yu, Gore Kao, Shuo-Han Chen, Yung-Hsiang Lu, George K. Thiruvathukal, and Ming Yin. 2020. Crowdsourcing detection of sampling biases in image datasets. In *Proceedings of the Web Conference*. 2955–2961.
- [59] Vasileios Iosifidis and Eirini Ntoutsi. 2018. Dealing with bias via data augmentation in supervised learning scenarios. *Jo Bates Paul D. Clough Robert Jäschke* 24 (2018).
- [60] Hosagrahar V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. 2014. Big data and its technical challenges. *Commun. ACM* 57, 7 (2014), 86–94.

- [61] Nikita Jaipuria, Xianling Zhang, Rohan Bhasin, Mayar Arafa, Punarjay Chakravarty, Shubham Shrivastava, Sagar Manglani, and Vidya N. Murali. 2020. Deflating dataset bias using synthetic data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 772–773.
- [62] 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 ACM SIGMOD International Conference on Management of Data*. 2721–2724.
- [63] Bo Kang, Jeffrey Lijffijt, and Tijl De Bie. 2018. Conditional network embeddings. *arXiv preprint arXiv:1805.07544* (2018).
- [64] Jian Kang and Hanghang Tong. 2021. Fair graph mining. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4849–4852.
- [65] Kimmo Karkkainen and Jungseock Joo. 2021. FairFace: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1548–1558.
- [66] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.
- [67] Ahmad Khajehnejad, Moein Khajehnejad, Mahmoudreza Babaei, Krishna P. Gummadi, Adrian Weller, and Baharan Mirzasoleiman. 2022. CrossWalk: Fairness-enhanced node representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11963–11970.
- [68] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A. Efros, and Antonio Torralba. 2012. Undoing the damage of dataset bias. In *Proceedings of the European Conference on Computer Vision*. Springer, 158–171.
- [69] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2016. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807* (2016).
- [70] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R. Rickford, Dan Jurafsky, and Sharad Goel. 2020. Racial disparities in automated speech recognition. *Proc. Nat. Acad. Sci.* 117, 14 (2020), 7684–7689.
- [71] Charlotte Laclau, Ievgen Redko, Manvi Choudhary, and Christine Largeron. 2021. All of the fairness for edge prediction with optimal transport. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. PMLR, 1774–1782.
- [72] Jennifer Langston. 2015. Who’s a CEO? Google image results can shift gender biases. Retrieved from <https://www.washington.edu/news/2015/04/09/whos-a-ceo-google-image-results-can-shift-gender-biases/>.
- [73] Alyssa Whitlock Lees and Ananth Balashankar. 2019. Fairness sample complexity and the case for human intervention. *Where is the Human? Bridging the Gap Between AI and HCI, CHI Workshop 2019*. <https://michae.lv/ai-hci-workshop/#call-for-participation>.
- [74] Joseph Lemley, Filip Jagodzinski, and Razvan Andonie. 2017. Big holes in big data: A Monte Carlo algorithm for detecting large hyper-rectangles in high dimensional data. *CoRR* abs/1704.00683 (2017).
- [75] Yi Li and Nuno Vasconcelos. 2019. Repair: Removing representation bias by dataset resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9572–9581.
- [76] M. Lichman. 2013. Adult Income Dataset, UCI Machine Learning Repository. Retrieved from <https://archive.ics.uci.edu/ml/datasets/adult>.
- [77] Yin Lin, Yifan Guan, Abolfazl Asudeh, and H. V. Jagadish. 2020. Identifying insufficient data coverage in databases with multiple relations. *Proc. VLDB Endow.* 13, 12 (2020), 2229–2242.
- [78] Bing Liu, Liang-Ping Ku, and Wynne Hsu. 1997. Discovering interesting holes in data. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI)*. Morgan Kaufmann Publishers Inc., 930–935.
- [79] Bing Liu, Ke Wang, Lai-Fun Mun, and Xin-Zhi Qi. 1998. Using decision tree induction for discovering holes in data. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (Lecture Notes in Computer Science, Vol. 1531)*. Springer, 182–193.
- [80] Chunxi Liu, Michael Picheny, Leda Sari, Pooja Chitkara, Alex Xiao, Xiaohui Zhang, Mark Chou, Andres Alvarado, Caner Hazirbas, and Yatharth Saraf. 2022. Towards measuring fairness in speech recognition: Casual conversations dataset transcriptions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6162–6166.
- [81] Li Lucy and David Bamman. 2021. Gender and representation bias in GPT-3 generated stories. In *Proceedings of the 3rd Workshop on Narrative Understanding*. 48–55.
- [82] Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W. Black. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047* (2019).
- [83] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Comput. Surv.* 54, 6 (2021), 1–35.
- [84] Michele Merler, Nalini Ratha, Rogerio S. Feris, and John R. Smith. 2019. Diversity in faces. *arXiv preprint arXiv:1901.10436* (2019).

- [85] Y. Moskovitch and H. Jagadish. 2021. Patterns count-based labels for datasets. In *Proceedings of the IEEE 37th International Conference on Data Engineering (ICDE)*. 1961–1966.
- [86] Yuval Moskovitch and H. V. Jagadish. 2020. COUNTATA: Dataset labeling using pattern counts. *Int. J. Very Large Data Bases* 13, 12 (2020), 2829–2832.
- [87] Yuval Moskovitch, Jinyang Li, and H. V. Jagadish. 2022. Bias analysis and mitigation in data-driven tools using provenance. In *Proceedings of the 14th International Workshop on the Theory and Practice of Provenance*. 1–4.
- [88] Fatemeh Nargesian, Abolfazl Asudeh, and H. V. Jagadish. 2021. Tailoring data source distributions for fairness-aware data integration. *Proc. VLDB Endow.* 14, 11 (2021), 2519–2532.
- [89] Fatemeh Nargesian, Abolfazl Asudeh, and H. V. Jagadish. 2022. Responsible data integration: Next-generation challenges. *Proceedings of the SIGMOD Conference*.
- [90] Mark E. J. Newman. 2003. Mixing patterns in networks. *Phys. Rev. E* 67, 2 (2003), 026126.
- [91] Jerzy Neyman and Egon Sharpe Pearson. 1967. Contributions to the theory of testing statistical hypotheses. In *Joint Statistical Papers*. University of California Press, Berkeley, 203–239. DOI : <https://doi.org/doi:10.1525/9780520339897-01>
- [92] Laura Niss, Yuekai Sun, and Ambuj Tewari. 2022. Achieving representative data via convex hull feasibility sampling algorithms. *arXiv preprint arXiv:2204.06664* (2022).
- [93] Eirini Ntoutsi, Pavlos Fafalios, Ujwal Gadiraju, Vasileios Iosifidis, Wolfgang Nejdl, Maria-Esther Vidal, Salvatore Ruggieri, Franco Turini, Symeon Papadopoulos, Emmanouil Krasnakis, et al. 2020. Bias in data-driven artificial intelligence systems' an introductory survey. *Wiley Interdisc. Rev.: Data Mining Knowl. Discov.* 10, 3 (2020), e1356.
- [94] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. 2019. Social data: Biases, methodological pitfalls, and ethical boundaries. *Front. Big Data* 2 (2019), 13.
- [95] Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. 2020. Bias in word embeddings. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 446–457.
- [96] Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. *arXiv preprint arXiv:1808.07231* (2018).
- [97] Eliana Pastor, Luca de Alfaro, and Elena Baralis. 2021. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the International Conference on Management of Data*. 1400–1412.
- [98] Dana Pessach and Erez Shmueli. 2022. A review on fairness in machine learning. *ACM Comput. Surv.* 55, 3 (2022), 1–44.
- [99] Romila Pradhan, Jiongli Zhu, Boris Glavic, and Babak Salimi. 2021. Interpretable data-based explanations for fairness debugging. *arXiv preprint arXiv:2112.09745* (2021).
- [100] Tahleen Rahman, Bartłomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. International Joint Conferences on Artificial Intelligence Organization, 3289–3295. DOI : <https://doi.org/10.24963/ijcai.2019/456>
- [101] Kumar Rakesh, Subhangi Dutta, and Kumara Shama. 2011. Gender recognition using speech processing techniques in LABVIEW. *Int. J. Adv. Eng. Technol.* 1, 2 (2011), 51.
- [102] Svetlana Sagadeeva and Matthias Boehm. 2021. SliceLine: Fast, linear-algebra-based slice finding for ML model debugging. In *Proceedings of the International Conference on Management of Data*. 2290–2299.
- [103] Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Gender bias in machine translation. *Trans. Assoc. Computat. Linguist.* 9 (2021), 845–874.
- [104] Nina Schaaf, Omar de Mitri, Hang Beom Kim, Alexander Windberger, and Marco F. Huber. 2021. Towards measuring bias in image classification. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 433–445.
- [105] Ben Schmidt. 2015. Rejecting the gender binary: A vector-space operation. *Bens Bookworm Blog* (2015). <http://bookworm.benschmidt.org/posts/2015-10-30-rejecting-the-gender-binary.html>.
- [106] L. S. Shapley. 1953. 17. A Value for n-Person Games. In *Contributions to the Theory of Games (AM-28)*, Harold William Kuhn and Albert William Tucker (Eds.). Vol. II. Princeton University Press, Princeton, 307–318. DOI : <https://doi.org/doi:10.1515/9781400881970-018>
- [107] Shubham Sharma, Yunfeng Zhang, Jesús M. Ríos Aliaga, Djallel Bouneffouf, Vinod Muthusamy, and Kush R. Varshney. 2020. Data augmentation for discrimination prevention and bias disambiguation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 358–364.
- [108] Shubhanshu Shekhar, Greg Fields, Mohammad Ghavamzadeh, and Tara Javidi. 2021. Adaptive sampling for minimax fair classification. *Adv. Neural Inf. Process. Syst.* 34 (2021), 24535–24544.
- [109] Suraj Shetiya, Ian P. Swift, Abolfazl Asudeh, and Gautam Das. 2022. Fairness-aware range queries for selecting unbiased data. In *Proceedings of the IEEE 38th International Conference on Data Engineering (ICDE)*.
- [110] Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. 2021. FairDrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Trans. Artif. Intell.* 3, 3 (2021), 344–354.

- [111] Julia Stoyanovich, Bill Howe, and H. V. Jagadish. 2020. Responsible data management. *Proc. VLDB Endow.* 13, 12 (2020).
- [112] Seymour Sudman. 1976. *Applied Sampling*. Technical Report. Academic Press, New York.
- [113] Chenkai Sun, Abolfazl Asudeh, H. V. Jagadish, Bill Howe, and Julia Stoyanovich. 2019. MithraLabel: Flexible dataset nutritional labels for responsible data science. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2893–2896.
- [114] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. *arXiv preprint arXiv:1906.08976* (2019).
- [115] Harini Suresh and John Guttag. 2021. A framework for understanding sources of harm throughout the machine learning life cycle. In *Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO'21)*. Association for Computing Machinery, New York, NY, USA, Article 17, 1–9. <https://doi.org/10.1145/3465416.3483305>
- [116] Ki Hyun Tae and Steven Euijong Whang. 2021. Slice tuner: A selective data acquisition framework for accurate and fair machine learning models. In *Proceedings of the International Conference on Management of Data*. 1771–1783.
- [117] Antonio Torralba and Alexei A. Efros. 2011. Unbiased look at dataset bias. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*. IEEE, 1521–1528.
- [118] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 11 (2008).
- [119] Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2019. Getting gender right in neural machine translation. *arXiv preprint arXiv:1909.05088* (2019).
- [120] Pranav Narayanan Venkit and Shomir Wilson. 2021. Identification of bias against people with disabilities in sentiment analysis and toxicity detection models. *arXiv preprint arXiv:2111.13259* (2021).
- [121] Angelina Wang, Arvind Narayanan, and Olga Russakovsky. 2020. REVISE: A tool for measuring and mitigating bias in visual datasets. In *Proceedings of the European Conference on Computer Vision*. Springer, 733–751.
- [122] Seyma Yucer, Samet Akçay, Noura Al-Moubayed, and Toby P. Breckon. 2020. Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 18–19.
- [123] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876* (2018).
- [124] Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. *arXiv preprint arXiv:1809.01496* (2018).

Received 15 March 2022; revised 25 September 2022; accepted 14 February 2023