Searching for Unfairness in Algorithms' Outputs: Novel Tests and Insights

Ian Davidson[1]*, **S. S. Ravi**[2]

[1] Department of Computer Science, University of California Davis, Davis, CA 95616
[2] Biocomplexity Institute, University of Virginia, Charlottesville, VA 22904
indavidson@ucdavis.edu, ssravi0@gmail.com

Abstract

As AI algorithms are deployed extensively, the need to ensure the fairness of their outputs is critical. Most existing work is on "fairness by design" approaches that incorporate limited tests for fairness into a limited number algorithms. Here, we explore a framework that removes these limitations and can be used with the output of any algorithm that allocates instances to one of K categories/classes such as outlier detection (OD), clustering and classification. The framework can encode standard and novel fairness types beyond simple counting, and importantly, it can detect intersectional unfairness without being specifically told what to look for. Our experimental results show that both standard and novel types of unfairness exist extensively in the outputs of fair-by-design algorithms and the counter-intuitive observation that they can actually increase intersectional unfairness.

1 Introduction and Motivation

The AI community has made tremendous progress towards ensuring fairness of algorithms with respect to groups determined by protected status variables (PSVs). Fairness here typically refers to disparate impact definitions of group-level fairness (Feldman et al. 2015). For example, suppose 50% of a population is female and 30% of the population is Hispanic. Then any algorithm that selects a subset of instances for an action (whether it is for a cluster, class or outlier group) should ensure that in the chosen subset, approximately 50% and 30% are women and Hispanic respectively. Adding such fairness criteria to existing algorithms has been well studied but typically does not guarantee intersectional fairness specifications such as "15% of the instances in a cluster should be Hispanic Women", unless specifically instructed to do so (Cho, Crenshaw, and McCall 2013). This is understandable: if there are m protected statuses, adding intersectional fairness requirements can generate $\Omega(2^m)$ constraints which quickly makes most ways of adding fairness by design to algorithms untenable. This is further compounded for protected statuses with k states (e.g., `race`) that are typically encoded as k binary protected statuses.

In this work, we view an algorithm as partitioning people into categories (e.g., classes, clusters, outlier/inlier) and auditing as determining whether some subgroup (determined by a combination of PSVs) is under-represented in any category. Although the previous work on auditing (Davidson and Ravi 2020a; Kearns et al. 2018) is useful, it is limited to (i) the two class setting, (ii) binary PSVs and (iii) unweighted fairness measures. Here unweighted means the utility/benefit of belonging to each category is the same. Binary protected status is particularly limiting; of the thirteen protected statuses in USA, only typically two (`sex-at-birth`, `citizenship`) are actually binary. Further, we show that though traditional fairness definitions can be satisfied, unfairness can occur when considering weights/utilities. While presenting the following motivating examples, we also discuss relevant related work.

A Motivating Example for Intersectional Unfairness Checking of Fair-By-Design Algorithms. The task of outlier detection (OD) (Hawkins 1980) is perhaps the most controversial application of deep learning as it is used to identify entities which are then policed (e.g., medical claims (Zhang and He 2017; Bauder and Khoshgoftaar 2017)), scrutinized (e.g., financial transactions (Huang et al. 2018; Zamini and Hasheminejad 2019)) or excluded (e.g., social media posting and account creation) (Yu et al. 2016; Savage et al. 2014)). Recent work (Shekhar, Shah, and Akoglu 2021; Zhang and Davidson 2021) has made strong progress towards increasing OD algorithms' fairness; this is achieved by including fairness as an additional part of the loss function. Perhaps the most compelling example of this work is the claim they make the OD algorithm's output fairer when applied to face data sets (Liu et al. 2015) as shown in Figure 1.

However, in our experiments we show that while these algorithms satisfy fairness for each group separately, they actually increase intersectional unfairness from 39% to 48% for two-way intersections and from 49% to 64% for three-way intersections. The number of unfairly treated subgroups for larger combinations is likely to be higher. Importantly, as this data set has 40 groups, changing the loss function to satisfy $2^{40}+$ combinations of intersectional unfairness would require that many additional loss function terms.

A Motivating Example for New Weighted Forms of Fairness. Consider the following situation which existing work cannot easily address. A credit card company divides its customer base into K categories and offers each category a different loyalty bonus. We would like each group (and sub-

*Corresponding Author

Figure 1: Earlier work (Zhang and Davidson 2021) claims that deep OD methods (normal group left, outlier group right) are inherently unfair (top row) as they overwhelming label people of color and men as outliers and that adding in fairness requirement fixes this problem (bottom row). However, applying our methods to their output shows they introduce new types of unfairness. This figure is reproduced from (Zhang and Davidson 2021).

group) to have in aggregate approximately equal bonuses. It is important to realize that it is possible to satisfy disparate impact requirements and still give some sub-groups substantially less bonuses if they are allocated to categories with lower bonuses too often. The classic two-class auditing work (Kearns et al. 2018; Davidson and Ravi 2020a) does not fit this setting even if applied multiple times. Further, this earlier work assumes just one desirable category but here each category's value varies.

Our Contributions: We make two styles of contributions. We provide formulations and approaches for checking for fairness and also experimental insights.

(1) We formulate the search for unfairness as a combinatorial optimization problem and establish its computational intractability (Theorem 1), leading to a test that cannot be easily side-stepped. Most importantly, even if the algorithm knows how it is being unfair, it is still intractable to pass our test of unfairness.

(2) We search for two types of unfairness: (a) count-based unfairness such as statistical parity and (b) a novel utility weighted (i.e., cost sensitive) unfairness which allows the benefits of some classes/actions to be more/less than others.

(3) For all formulations, our methods allow finding unfairness across multiple combinations of PSVs. We experimentally show that fair-by-design OD and clustering algorithms can actually increase intersectional unfairness.

2 A High Level Overview of Our Tests

We begin with a basic formulation that is similar to the classical count-based methods in the literature (e.g., (Chierichetti et al. 2017)) and then extend it to a weighted formulation. Other formulations are possible so long as fairness can be encoded as a constraint.

Our work searches for an under-represented PSV combination denoted by a binary vector $\mathbf{x}$ (which represents a subgroup of individuals) in a single category produced by the algorithm. If no such PSV combination is returned for all categories, then we conclude that the division of people into categories is fair. If a PSV combination is returned, then the domain expert can determine whether the type of unfairness found is acceptable (or uninteresting), and our formulations can be run again to explicitly avoid finding such examples of unfairness. To tie our work back to the Set Cover (Garey and Johnson 1979) problem in theoretical computer science, we formulate our work as searching for either a conjunction or disjunction of PSVs (e.g., $\texttt{Hispanic} \wedge \texttt{Female}$, $\texttt{Female} \vee \texttt{Elderly}$) that is under-represented in a class compared to other classes (e.g., the rest of the population). We show a complexity result for the disjunctive case and quadratic integer programming (QIP) formulations for the conjunctive case.

Types of unfairness considered. The two types of unfairness considered in our work are outlined below.

(1) *Count-based unfairness.* This uses a test similar to the traditional definition of statistical parity (Kearns et al. 2018); i.e., the count of instances in a class satisfying a PSV combination $\mathbf{x}$ (normalized by the class size) differs significantly from the proportion of $\mathbf{x}$ in the population. Other count-based definitions of fairness such as equalized odds and predictive parity can be encoded as counting constraints.

(2) *Utility-weighted (or Cost-sensitive) unfairness.* Here, rather than one category being desirable, each of the K categories may have a different utility/cost $(U_1, \ldots, U_K)$. Lemma 1 calculates the expected utility when sub-groups are randomly allocated to categories. Our tests compare the actual utility to this expected amount to identify unfairness.

3 Formulations of Unfairness

In this section, we develop rigorous formulations of the two forms of unfairness studied in this paper. The notation used in these formulations is summarized in Table 1. The reader can understand our tests for unfairness by looking at Problems 1 and 2. The rest of the section details how they are implemented as quadratic integer programming (QIP) problems for experimentation and reproducibility.

3.1 Count-Based Unfairness

Here, we outline our test of count-based unfairness for one class (the target category/action/class) which is repeated K times, where K is the number of categories (or actions/-classes), with each category taking a turn at being the target class. It is important to understand that our test is formulated as a search problem with the aim of finding a **simplest**[1] **example of unfairness**. We choose to find the simplest PSV combination as it indicates the most general form of unfairness. If there is no solution for any class, the algorithm's output is deemed fair.

[1] We use "simplest" to mean a vector $\mathbf{x}$ with the smallest number of PSVs.

Variable	Meaning
$\mathbb{P},\ m$	The set and the number of PSVs (i.e., $m = \|\mathbb{P}\|$).
$\mathbf{x}$	Binary selection vector for the PSVs of subgroups. (Each vector represents a subset of $\mathbb{P}$.)
$\mathbb{T},\ \mathbb{O},\ \mathbb{C}_k$	The set of instances in a target class, other class and the k^{th} class respectively. (We let $r = \|\mathbb{T}\|$ and $t_k = \|\mathbb{O}_k\|$.)
$\mathbb{T}^j,\ \mathbb{O}_k^j,\ \mathbb{C}_k^j$	The PSV vectors for the j^{th} instance in the target class, the k^{th} other class and class k respectively.
$y_k^j,\ z_k^j$	Indicator variables for the j^{th} instance in class k. The variable y_k^j (z_k^j) is 1 iff the j^{th} instance in class k is covered by $\mathbf{x}$. (Note: In the UDSC problem, z is used to cover only the target class.)
$U_1, U_2, \ldots, U_K$	Utility (benefit) values associated with classes $\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_K$ respectively.
$\alpha,\ \beta, \lambda, \mu$	Coverage and utility bounds with $\alpha < \beta$ and $\lambda < \mu$. Values $\beta - \alpha$ and $\mu - \lambda$ specify the tolerance to unfairness.
$k,\ K$	An index to classes and the total number of classes respectively.
$a_k,\ b_k$	Lower and upper bounds on the utility of the k^{th} class, $1 \leq k \leq K$.

Table 1: List of variables used in the mathematical programming formulations developed in the paper.

Problem Definition. The objective of our optimization problems is shown diagrammatically in Figure 2. The figure shows K Venn diagrams (one for each class), and the coverage of the subgroup ($\mathbf{x}$) with respect to the PSVs is denoted by a black dashed rectangle. A formal definition of coverage for conjunctions of PSVs is as follows.

Definition 1. *Let $\mathbb{C}$ be a class and let vector $\boldsymbol{x}$ represent a subset of (binary valued) PSVs. The set of instances in $\mathbb{C}$ **covered** by $\boldsymbol{x}$ includes each instance η in $\mathbb{C}$ such that η possesses all PSVs in $\boldsymbol{x}$.*

For a disjunction of PSVs, the definition of coverage is similar, except that the phrase "all PSVs" is replaced by "at least one PSV".

Problem 1. ***Unfairness Detection In a Single (Target) Class (UDSC) Problem.*** *A decision version of this problem can be expressed formally as follows:*

$$\exists \boldsymbol{x} : P(\boldsymbol{x}|\mathbb{T}) \leq \alpha,\ \ P(\boldsymbol{x}|\neg\mathbb{T}) \geq \beta \ \text{ and } \ \alpha < \beta,$$

where $P(\boldsymbol{x}|\mathbb{T})$ and $P(\boldsymbol{x}|\neg\mathbb{T})$ indicate respectively the fraction of instances in $\mathbb{T}$ and not in $\mathbb{T}$ covered by $\boldsymbol{x}$.

Example: Suppose we have three binary PSVs, {`Female`, `LowIncome`, `Married`}, and we search for conjunctions with $\alpha = 0.1$ and $\beta = 0.2$. If our method finds $\mathbf{x}$ = (`True`, `True`, `False`) this shows that this subgroup exists with chance at most 0.1 in one class and chance at least 0.2 in all others. This means the sub-group {`Female` $\wedge$ `LowIncome`} is less than half as likely to occur in the target class $\mathbb{T}$ compared to other classes.

As the optimization problem's goal is to find a smallest (hence simplest) subgroup ($\mathbf{x}$), we find the most general under-represented subgroup.

A Quadratic Integer Program (QIP) for Detecting Unfairness in One Class. This section can be skipped on first reading with the understanding it defines a mathematical programming formulation for Problem 1.

Given K classes, we wish to find whether there is a PSV conjunction which is under-represented in any class. We search for a subset of PSVs as given by the binary indicator vector $\mathbf{x}$. Let $q = ||\mathbf{x}||$ be the number of PSVs selected. For convenience, let $t_k = |\mathbb{O}_k|$ be the size of the k^{th} class, $1 \leq k \leq K$. We compute the *fraction* of instances in $\mathbb{T}$ (i.e., the target class) and $\mathbb{O}_1, \ldots, \mathbb{O}_{K-1}$ (i.e., the other $K-1$ classes) that are covered by $\mathbf{x}$. To do this through an QIP, we represent the PSV vector of the j^{th} instance in the k^{th} class by $\mathbb{O}_k^j$. Similarly, the i^{th} instance's PSV vector in the target class is denoted by $\mathbb{T}^i$. Then to compute the fraction of instances in $\mathbb{T}$ covered by $\mathbf{x}$, we introduce binary variables z_1, z_2, …, z_r, where $r = |\mathbb{T}|$. We ensure that $z_i = 1$ iff the vector $\mathbf{x}$ covers the i^{th} point in $\mathbb{T}$. Thus, $\sum_{i=1}^{r} z_i$ gives the number of instances in $\mathbb{T}$ covered by $\mathbf{x}$. We want $\mathbf{x}$ to cover at most α fraction of the points in $\mathbb{T}$.

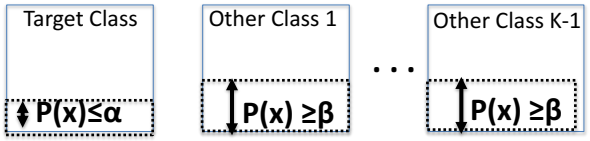

Figure 2: Our unfairness detection problem to find a subgroup (denoted by $\mathbf{x}$) that is under-represented in one class (occurs in less than α fraction of instances) and relatively over-represented in the others (occurs in more than β fraction of instances). The value $\gamma = \beta - \alpha$ is the *disparity gap*, which is a probability for count-based unfairness and a numerical value for utility-based unfairness.

For each class $\mathbb{O}_k$ ($1 \leq k \leq K-1$), we use $t_k = |\mathbb{O}_k|$ additional 0/1 variables, denoted by $y_k^1, y_k^2, \ldots, y_k^{t_k}$; here, variable y_k^j corresponds to the j^{th} instance in class $\mathbb{O}_k$. We add constraints so that $y_k^j = 1$ iff the j^{th} instance in $\mathbb{O}_k$ contains all the PSVs in $\mathbf{x}$. Hence, $\sum_{j=1}^{t_k} y_k^j$ gives the number of instances in $\mathbb{O}_k$ covered by the vector $\mathbf{x}$. We add constraints to ensure that at least β fraction of instances in **each** of the classes $\mathbb{O}_1, \ldots, \mathbb{O}_{K-1}$ are covered by $\mathbf{x}$.

If we set $\alpha = 0.5\beta$ and a solution to our optimization problem is found, then $\mathbf{x}$ contains a PSV combination that matches a subgroup that is under-represented in $\mathbb{T}$ and over-represented in all of the classes $\mathbb{O}_1, \ldots, \mathbb{O}_{K-1}$ by a factor of 2. Conversely, if no solution is found, then no such unfairness exists (given the requirements set by α and β). Based on the above discussion, we formulate the following QIP for

the optimization version of the UDSC problem.

QIP formulation for Problem 1:

Objective: $\text{argmin}_{\mathbf{x},\mathbf{y},\mathbf{z}} \; ||\mathbf{x}||$

Constraints: Let $q = ||\mathbf{x}||$.

(i) $q\,y_k^j \leq \mathbf{x}^T\,\mathbb{O}_k^j, \;\; 1 \leq j \leq t_k.$

(ii) $q\,y_k^j \geq \mathbf{x}^T\,\mathbb{O}_k^j - (q-1), \;\; 1 \leq j \leq t_k.$

(iii) $q\,z_i \leq \mathbf{x}^T\,\mathbb{T}^i, \;\; 1 \leq i \leq |\mathbb{T}|.$

(iv) $q\,z_i \geq \mathbf{x}^T\,\mathbb{T}^i - (q-1), \;\; 1 \leq i \leq |\mathbb{T}|.$

(v) $\sum_{i=1}^{r} z_i \leq \alpha|\mathbb{T}|.$

(vi) $\sum_{j=1}^{t_k} y_k^j \geq \beta|\mathbb{O}_k|, \;\; 1 \leq k \leq K-1.$

Notes about the QIP: Our objective finds a smallest combination of PSVs for a subgroup treated unfairly. All the variables (represented by $\mathbf{x}$, y and z) take on values from $\{0,1\}$. The first two constraints ensure that each binary variable y_k^j is set to 1 iff instance j in class k has the PSV combination $\mathbf{x}$. The next two constraints are as above but for the target class. The final two constraints count and check how many times $\mathbf{x}$ occurs in the target class and other classes.

3.2 Utility-Weighted Unfairness

Our formulation in Problem 1 implicitly gives each class/action an equal weight in terms of how desirable or undesirable it is. Here we allow these weights ("utilities") to be either given as constants or be solved for within given bounds. This is compared with the expected utility for the instances being randomly distributed across the classes/actions which we discuss next.

Lemma 1. *Let U_k denote the utility assigned to class k, $1 \leq k \leq K$. Suppose the instances covered by a PSV combination $\boldsymbol{x}$ are distributed uniformly randomly over the K classes. Then the total expected utility of the instances covered by $\boldsymbol{x}$ is $(N_{\boldsymbol{x}}/K)\sum_{k=1}^{K} U_k$, where $N_{\boldsymbol{x}}$ is the number of instances covered by $\boldsymbol{x}$ in the population.*

Proof: Let $\ell = N_{\mathbf{x}}$ and $M = \{w_1, w_2, \ldots, w_\ell\}$ be the set of all instances covered by $\mathbf{x}$. Let h_i be the random variable (RV) that gives the utility of w_i when the instances in M are distributed uniformly randomly across the K classes, $1 \leq i \leq \ell$. Thus, the RV $H = \sum_{i=1}^{\ell} h_i$ gives the total utility of the instances in M. By linearity of expectation (Mitzenmacher and Upfal 2005), $E[H] = \sum_{i=1}^{\ell} E[h_i]$. To find $E[h_i]$, note that the probability that w_i is assigned to a specific class k is $1/K$ and the corresponding utility is U_k. Thus, $E[h_i] = \sum_{k=1}^{K} U_k/K = (1/K)\sum_{k=1}^{K} U_k$. Hence, $E[H] = \sum_{i=1}^{\ell} E[h_i] = (\ell/K)\sum_{k=1}^{K} U_k$. Since $\ell = N_{\mathbf{x}}$, the lemma follows. ■

The above lemma can be used to test if the actual utilities (from an algorithm's output) differ significantly from the expected values.

Problem 2. ***Utility Weighted-Unfairness Detection.***

$\exists U, \boldsymbol{x} : [\sum_k |\mathbb{C}_k| P(\boldsymbol{x}|\mathbb{C}_k) U_k] \leq \lambda$ (i.e., Actual Utility $\leq \lambda$), $(N_{\boldsymbol{x}}/K)\sum_k U_k \;\geq\; \mu$ (i.e., Expected Utility $\geq \mu$, see Lemma 1), $\lambda < \mu$ and $a_k \leq U_k \leq b_k, \;\; 1 \leq k \leq K$, *where $N_{\boldsymbol{x}}$ is the number of instances covered by $\boldsymbol{x}$ in the population and $P(\boldsymbol{x}|\mathbb{C}_k)$ is the fraction of instances in $\mathbb{C}_k$ covered by $\boldsymbol{x}$.*

The above problem states that the actual weighted utility across all classes should be $\leq \lambda$ and the expected utility for the uniform distribution across the K classes should be $\geq \mu$.

QIP Formulation for Problem 2:

Objective: $\text{argmin}_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{U}} \; ||\mathbf{x}||$

Constraints: Let $q = ||\mathbf{x}||$.

(i) $q\,y_k^j \leq \mathbf{x}^T\,\mathbb{C}_k^j, \;\; 1 \leq j \leq t_k.$

(ii) $q\,y_k^j \geq \mathbf{x}^T\,\mathbb{C}_k^j - (q-1), \;\; 1 \leq j \leq t_k.$

(iii) $\sum_{k=1}^{K} \left(U_k \sum_{j=1}^{t_k} y_k^j\right) \leq \lambda.$

(iv) $(N_{\mathbf{x}}/K)\sum_{k=1}^{K} U_k \geq \mu.$

(v) $a_k \leq U_k \leq b_k, \;\; 1 \leq k \leq K.$

In specifying the constraints, we used $N_{\mathbf{x}}$ for simplicity; in terms of the variables, $N_{\mathbf{x}}$ is equal to $\sum_{k=1}^{K}\sum_{j=1}^{t_k} y_k^j$. The above formulation is similar to that for Problem 1, except that it is more complex as the constraints now are weighted by the cluster utilities.

4 Experiments

We illustrate the practicality[2] of our tests for the outputs of: (a) fair OD algorithms, (b) fair clustering algorithms, and (c) a human process (namely, electoral map creation).

4.1 Intersectional Unfairness in Outlier Detection

A key source of "machine bias" (ProPublica 2016) is unrepresentative (training) data which can also occur in OD. The sample size of a minority subpopulation, by definition, is smaller than that of the majority group. Outlier detectors are designed *exactly* to spot such rare, minority samples and increase their chance of getting audited/"policed" or otherwise filtered.

Recent works attempt to correct these issues and claim to address fairness across multiple groups (Shekhar, Shah, and Akoglu 2021; Song, Li, and Liu 2021; Zhang and Davidson 2021). Here we audit the output of one work (Zhang and Davidson 2021) and show that despite these claims, there is considerable intersectional unfairness. With the CelebA data set (Liu et al. 2015), there are forty groups and we find that applying this earlier work to all groups actually yields more intersectional unfairness than if it weren't applied.

Celeb-A Data Set. This is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations such `race`, `gender`, `skin color` though some annotations are not protected status (e.g., `chubby`, `eye-glasses`). We used the standard 50,000 instance subset of this data set and treated all forty annotations as groups. This leads to far more than 40 binary protected statuses as many annotations are multi-state (e.g., `race` takes on many values).

Deep Fair OD Algorithm (Zhang and Davidson 2021). We replicate this earlier work that claims to address fairness

[2] All the QIP formulations were implemented in Gurobi (Gurobi Optimization, LLC 2023). All data sets used are publicly available.

for multiple groups which we summarize below. Their work uses the deep SVDD network (Ruff et al. 2018) that tries to learn a function $f(g(x))$ that maps all points close to a predefined centroid c. Here, $g(x)$ is an encoding network. Formally, the objective is $\text{argmin}_{f,g}||f(g(x))-c||$, with a natural outlier score $s(x) = ||f(g(x))-c||$, that is the Euclidean (or some other) distance. To this objective is added the aim to learn a classifier h to predict poorly the protected status (group) y but using the same encoder $g(x)$. Formally:

$$\text{argmin}_{f,g,h}||f(g(x)-c|| - |h(g(x))-y|\,. \qquad (1)$$

The above formulation is for one group, and in our experiment there are sixty-eight classification functions $(h_1, h_2, \ldots, h_{68})$.

Measuring Unfairness. Using the traditional measure of balance $min[min(\frac{c_o}{\neg c_o}, \frac{\neg c_o}{c_o}), min(\frac{c_n}{\neg c_n}, \frac{\neg c_n}{c_n})]$ (Chierichetti et al. 2017), we measure how fair or balanced the categories (outlier and normal) are. Here c_o ($\neg c_o$) is the number of people of protected status (not in protected status) in the outlier category and c_n ($\neg c_n$) the number of people of protected status (not in protected status) in the normal category. The ideal balance will depend on how abundant the protected group is in the population; for example, for a protected group occurring in 50% of the population, the perfect balance is 1.0. Following the standard rules of disparate impact (Song, Li, and Liu 2021), we say the group/sub-group is treated unfairly if the balance is more then 20% from the ideal.

Figure 3 shows the distributions of balance across all groups and shows no indication of disparate impact as would be expected since the algorithm aims to achieve this.

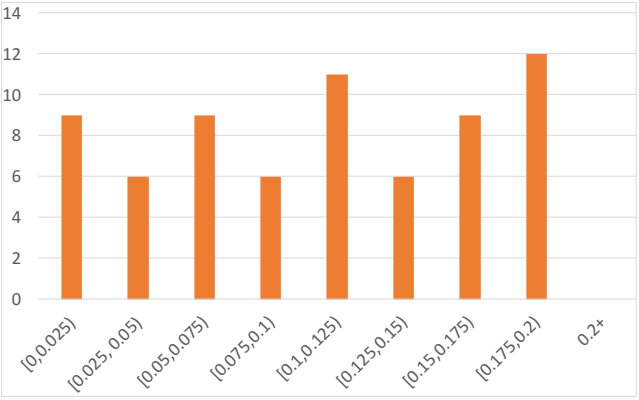

Figure 3: A histogram of the percentage deviation from the perfect balance (x-axis) versus frequency (y-axis). Note that there are sixty-eight (not forty) entries as non-binary groups (i.e., race) are encoded as multiple binary groups.

Search for Count-Based Intersectional Unfairness. Here we apply our first formulation to determine if there are any examples of intersectional unfairness in the output of their algorithm. Our method finds the simplest form of explanation which is then included as an orthogonality constraint (so as to not find it again) in subsequent applications. This effectively first finds examples of unfairness at the intersection of two groups. We repeat these experiments for three groups which took sixteen hours of computation time. We count the number of examples of unfairness and the most extreme forms of unfairness in Table 2. Importantly, using the base algorithm SVDD yielded far less intersectional unfairness than using the fair variant of SVDD presented in (Zhang and Davidson 2021).

Search for Utility-Based Intersectional Unfairness. Though many definitions of fairness exist such as statistical parity, equalized odds, etc. (Shekhar, Shah, and Akoglu 2021), they all have in common that one class is desirable and the other is not. But this is not often the case; many times in OD, the cases are divided into categories with each category getting a different degree of scrutiny (Hawkins 1980). For example, if outliers are transactions to investigate, then those towards the top of the list (i.e., a higher outlier score) are the most promising leads and get more investigation time than those further down the list (Huang et al. 2018).

To simulate this, we take the output of the OD algorithm and decile the outliers according to their outlier scores giving each instance a score of 1 to 10. Outliers with a score of 10 (the least unusual) are given 1 time unit of scrutiny, those with a score of 9 are given 2 time units, those with a score of 8 are given 3 times units and so forth until those with a score of 1 get 10 times as much scrutiny as those of score 10. Now consider a situation where two groups have an equal number of their members considered outliers, but one group's members are all in decile 10 and the other group's members are all in decile 1. Clearly, the latter will be more scrutinized but most counting measures (since they are unweighted) will consider them as equally treated.

We repeated the previous experiments for OD which are reported in Table 3 but find unfairness using Problem 2 and the utility values $U_1 = 1$, $U_2 = 2$, $U_3 = 3$, …, $U_{10} = 10$. Here, we say an algorithm's output is unfair with respect to a sub-group if its utility is 20% more (or less) than the expected amount given by Lemma 1. We find that the weighted intersectional groups of unfairness are often specializations of the regular unfairness. For example, the tag `Black` is replaced by `Big-Lips`, `Asian` is replaced by `Narrow-Eyes` and `Hispanic` gets replaced by `Wavy-Hair`.

4.2 Evaluating the Unfairness of Fair-By-Design Clustering Algorithms

We take the output of a classic (fairlet-based) fair-by-design clustering algorithm (Backurs et al. 2019) which ensures fairness for just one PSV and then measure fairness across the remaining PSVs. Even though this is a simple experiment, we believe that it is necessary to show these algorithms do not enforce intersectional (i.e., subgroup) fairness. We take the classic Adult data set (Dua and Graff 2017) studied by many fair clustering papers (Chierichetti et al. 2017; Bera, Chakrabarty, and Negahbani 2019; Davidson and Ravi 2020b; Schmidt, Schwiegelshohn, and Sholer 2018; Kleindessner, Awasthi, and Morgenstern 2019). This data set contains four PSVs: `gender`, `education`, `marital-status`, `occupation`. We produce a fair clustering for just one PSV (as the fairlets method allows)

No. of Intersections	% Unfair Intersections	Five Examples Most Different From Ideal Balance
Size 2: 1560	48%	[R:Black,Male],[R:Asian,Male],[R:Hispanic,Male],[Black Hair, Asian], [Black Hair, R:Black]
Size 3: 59280	64%	[R:Black,Male,Black Hair],[R:Asian,Male,Black Hair],[R:Hispanic,Male,Bushy Eyebrows], [R:Hispanic,Male,Goatee],[R:Black,Male,Eye-Glasses]

Table 2: Checking the count-based fairness of the output of the Fair SVDD Algorithm (Zhang and Davidson 2021). The number of examples of intersectional unfairness for 2 and 3 group combinations found using **QIP formulation for Problem 1**. The percentage of unfair intersections found using regular SVDD is Size 2: 39% and Size 3: 49%. Note intersections between mutually exclusive groups (e.g., Blond Hair, Black Hair) were excluded. Compare with Table 3.

No. of Intersections	% Unfair Intersections	Five Examples Most Different From Ideal Balance
Size 2: 1560	59%	[R:Narrow-Eyes,Male],[R:Big-Lips,Male],[R:Wavy-Hair, Male], [Wearing-Hat, Asian], [Wearing-Hat, R:Black]
Size 3: 59280	78%	[R:Narrow-Eyes,Male,Black Hair],[R:Wavy-Hair,Male,Black Hair],[R:Hispanic,Male,Goatee] [R:Hispanic,Male, Eye-Glasses], [R:Black,Heavy-Makeup,Eye-Glasses]

Table 3: Checking the utility (i.e., cost sensitive) fairness of the output of the Fair SVDD Algorithm (Zhang and Davidson 2021). The number of examples of weighted intersectional unfairness for 2 and 3 group combinations found using **QIP formulation for Problem 2**. The percentage of unfair intersections found using regular SVDD is Size 2: 37% and Size 3: 53% Note intersections between mutually exclusive groups (e.g., Blond Hair, Black Hair) were excluded. Compare with Table 2.

and then measure unfairness across the remaining three PSVs. In all experiments, we use $K = 6$ as is typical with this data set. This is done by solving the QIP for Problem 1 for each cluster in turn as the target; if any solution is returned, the clustering is deemed unfair and the PSV combination ($\mathbf{x}$) causing the unfairness noted. If a PSV combination is found, we re-run the QIP again with an additional orthogonality constraint to find a new PSV combination (example of unfairness) until no unfairness is discovered.

We set γ to be 20% less than the median population probability (mean of two middle values) of all PSV combinations. Our results (Table 4) indicate the need for measuring intersectional unfairness.

PSV Balanced	Number of Unfair Combinations in the Remaining PSVs
Gender (G)	4 (E, M, EM, O, OM)
Education (E)	2 (GO, GM)
Marital Status (M)	5 (E, EO, G, GO, EG)
Occupation (O)	2 (EM,MG)

Table 4: Finding intersectional unfairness in the output of a classic fairness-by-design clustering algorithms (Backurs et al. 2019) on the Adult data set. The algorithm balanced the PSV in the left column. We report the number and examples of intersectional unfairness in the remaining three PSVs (maximum = 8).

4.3 Utility-Based Unfairness for Electoral Maps[3]

The experiment in the previous subsection used our first formulation which treated all classes as being equally desirable. Here, we consider the situation where utility of the classes/clusters/actions ($U_1, \ldots, U_K$) are different. If a subgroup

[3]This section uses datasets from (California Data 2024).

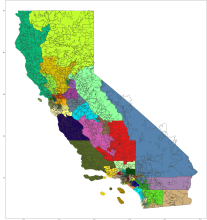

Figure 4: The 53 California congressional districts (the classes) and the 1700+ ZCTA (zip code tabulated areas) that comprise them (the instances).

(denoted by $\mathbf{x}$) has a measured utility for a set of classes ($C_1, \ldots, C_k$) that is substantially (+/-20%) different from the expected utility (over random allocation to classes), then the classification/clustering is deemed unfair.

California consists of 53 congressional districts (CDs) which can be considered categories. Each CD can be considered a class containing a subset of the 1700+ Zip Code Tabulation Areas (ZCTAs) (Bureau 2010) as shown in Figure 4. For each ZCTA, we have its assignment to a CD, population size and the fraction of its population having the following demographic attributes (Grubesic and Matisziw 2006): Foreign born, Chinese, Black, Indian, Vietnamese, Filipino, White, 65+ old, Female, Japanese, American Indian, Islander, and Native Hawaiian . Hence for each CD (category) we can aggregate the population and demographics of the people who live their. Further, each CD has a different median local property tax basis (per capita) which is used as the utility measure as it indicates a general quality of living, since property taxes typically fund schools, local sports, parks and other important quality of living indicators.

We use the QIP formulation for Problem 2. If no solution is found then the CDs are "fair" in that no PSV-combination

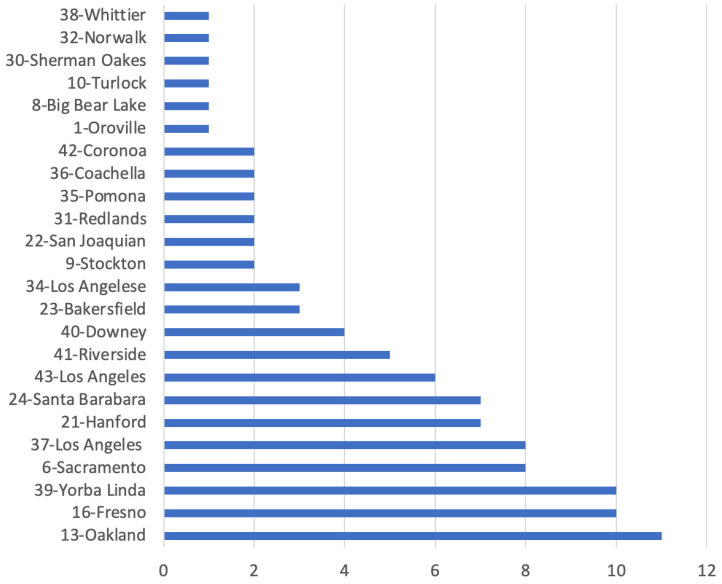

Figure 5: The distribution of 100 simplest forms of of unfairness across California's 53 congressional districts (CD). The y-axis refers to the CD and the x-axis shows how often unfairness was found in the CD. Districts not shown had no unfairness detected.

defined a subgroup of people allocated 20% less money than their expected utility if they were assigned randomly to the CDs. Our method finds the simplest forms of unfairness and we repeat our experiment 100 times, each time adding an orthogonality constraint to not find a previously found form of unfairness. We calculated the distribution of unfairness found in the 53 CDs and noticed that it is concentrated in the following districts: 13th-Oakland, 16th Fresno, 21st Hanford, 24th-Santa Barbara, 37th-Los Angeles and 39th La Habra (see Figure 5). Overwhelmingly, unfairness centered on race but not on country of birth or gender.

5 Complexity of Detecting Unfairness

We now show that count-based unfairness detection is **NP**-hard for disjunctions of PSVs. Our QIPs search for conjunction of PSVs. We conjecture that the version for conjunctions is also **NP**-hard (like other problems modeled by QIPs (Garey and Johnson 1979)), but leave it to future work. We start with a formal definition of the decision version of the problem for proving **NP**-hardness.

Unfairness Detection in a Single Class (UDSC)
Given: A collection of $K \geq 2$ pairwise disjoint classes $\mathbb{T}$, $\mathbb{O}_1$, …, $\mathbb{O}_{K-1}$ and a set $\mathbb{P} = \{p_1, p_2, \ldots, p_m\}$ of m PSVs, positive integers α and β, where $\alpha < \beta$.
Question: Is there a subset $P' \subseteq \mathbb{P}$ such that P' covers at most α instances of $\mathbb{T}$ and at least β instances in each of the other classes $\mathbb{O}_1$, $\mathbb{O}_2$, …, $\mathbb{O}_{K-1}$?

For simplicity in the proof we have α and β as integers. It is straightforward to express them as fractions of the population size. The UDSC problem defined above does not require one to minimize the number of PSVs. Nevertheless, we have the following result.

Theorem 1. UDSC *is* ***NP****-hard even for two classes.*

Proof (idea): Our reduction is from the **Minimum Set Cover** (MSC) problem: given a universe $U = \{u_1, u_2, \ldots, u_n\}$, a collection $S = \{S_1, S_2, \ldots, S_m\}$, where each $S_j \subseteq U$ $(1 \leq j \leq m)$ and an integer $r \leq m$, is there is a subcollection $S' \subseteq S$ such that $|S'| \leq r$ and the union of the sets in S' is equal to U? MSC is **NP**-complete even when $r < n$ (Garey and Johnson 1979). Our reduction produces two clusters, i.e., $\mathbb{T}$ (the target) and another cluster $\mathbb{O}$. For details, see (Davidson and Ravi 2022). ■

6 Summary, Limitations and Future Work

We explore new approaches for testing whether the output of an algorithm is fair as a series of combinatorial optimization problems designed to search for unfairness. Our first formulation uses a count-based definition of fairness and our second formulation uses utilities to model cost-sensitive unfairness across multiple classes. Since our formulations lead to **NP**-hard problems, they cannot be easily side-stepped even if one knows why the output is unfair. Our experimental work showed that fair-by-design algorithms for outlier detection and clustering introduce significant intersectional unfairness. Further, we demonstrated the flexibility of our work by showing that it can audit the output of human processes in electoral map creation.

One limitation of our work is that the formulations are for algorithms (classification, clustering, outlier detection) whose decisions induce a set partition on instances. A second limitation is that the approach needs an expert's guidance in choosing tolerance values to decide fairness. Since our methods search for the simplest examples of unfairness, the QIP formulations may be computationally expensive when the number of PSVs is large. Developing other formulations (e.g., relaxing the requirement of simplest examples) that can scale to a large number of PSVs is a topic for future research.

Ethical Statement

The work reported in this paper addresses fairness issues in the outputs generated by algorithms. The paper presents methods that are useful in checking whether results produced by algorithms may exhibit unfairness. The experimental results in the paper use public domain datasets.

Acknowledgments

We thank the AAAI 2025 Alignment Track referees for providing very helpful feedback. This work was supported in part by NSF Grants IIS-1908530 and IIS-1910306 titled "Explaining Unsupervised Learning: Combinatorial Optimization Formulations, Methods and Applications" and IIS-2310481 titled "Towards Fair Outlier Detection".

References

Backurs, A.; Indyk, P.; Onak, K.; Schieber, B.; Vakilian, A.; and Wagner, T. 2019. Scalable Fair Clustering. In *Proc. ICML*, 405–413. Online: PMLR.

Bauder, R. A.; and Khoshgoftaar, T. M. 2017. Multivariate anomaly detection in medicare using model residuals and probabilistic programming. *The Thirtieth International Flairs Conference*.

Bera, S. K.; Chakrabarty, D.; and Negahbani, M. 2019. Fair Algorithms for Clustering. Arxiv: abs/1901.02393v1.

Bureau, C. 2010. US Census Data. https://www.census.gov/tiger/tms/gazetteer/zcta5.txt.

California Data. 2024. Various Census Datasets. https://www.census.gov/data/datasets.html.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair Clustering Through Fairlets. In *Proc. NeurIPS*, 5036–5044. Red Hook, NY: Curran Associates, Inc.

Cho, S.; Crenshaw, K. W.; and McCall, L. 2013. Toward a field of intersectionality studies: Theory, applications, and praxis. *Signs: Journal of women in culture and society*, 38(4): 785–810.

Davidson, I.; and Ravi, S. S. 2020a. A Framework for Determining the Fairness of Outlier Detection. In *Proc. ECAI 2020*, 2465–2472.

Davidson, I.; and Ravi, S. S. 2020b. Making Existing Clusterings Fairer: Algorithms, Complexity Results and Insights. In *Proc. AAAI 2020*, 3733–3740. Online: AAAI Press.

Davidson, I.; and Ravi, S. S. 2022. Towards Auditing Unsupervised Learning Algorithms and Human Processes For Fairness. ArXiv Report: arXiv:2209.11762 [cs.AI].

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository.

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and Removing Disparate Impact. In *Proc. 21st ACM SIGKDD*, 259–268.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman & Co.

Grubesic, T. H.; and Matisziw, T. C. 2006. On the use of ZIP code tabulation areas (ZCTAs) for the spatial analysis of epidemiological data. *International journal of health geographics*, 5(1): 1–58.

Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual.

Hawkins, D. M. 1980. *Identification of outliers*, volume 11. New York, NY: Springer.

Huang, D.; Mu, D.; Yang, L.; and Cai, X. 2018. CoDetect: Financial Fraud Detection With Anomaly Feature Detection. *IEEE Access*, 6: 19161–19174.

Kearns, M.; Neel, S.; Roth, A.; and Wu, Z. S. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proc. ICML*, 2564–2572. Online: PMLR.

Kleindessner, M.; Awasthi, P.; and Morgenstern, J. 2019. Fair k-Center Clustering for Data Summarization. In *Proc. ICML*, 3448–3457. Online: PMLR.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep Learning Face Attributes in the Wild. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Mitzenmacher, M.; and Upfal, E. 2005. *Probability and Compting: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. New York, NY: Cambridge University Press.

ProPublica. 2016. Machine Bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S. A.; Binder, A.; Müller, E.; and Kloft, M. 2018. Deep one-class classification. In *International conference on machine learning*, 4393–4402. PMLR.

Savage, D.; Zhang, X.; Yu, X.; Chou, P.; and Wang, Q. 2014. Anomaly detection in online social networks. *Social networks*, 39: 62–70.

Schmidt, M.; Schwiegelshohn, C.; and Sholer, C. 2018. Fair Coresets and Streaming Algorithms for Fair k-Means Clustering. Arxiv: abs/1812.10854v1.

Shekhar, S.; Shah, N.; and Akoglu, L. 2021. Fairod: Fairness-aware outlier detection. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 210–220.

Song, H.; Li, P.; and Liu, H. 2021. Deep Clustering based Fair Outlier Detection. In *Proc. 27th ACM SIGKDD*, 1481–1489.

Yu, R.; Qiu, H.; Wen, Z.; Lin, C.; and Liu, Y. 2016. A survey on social media anomaly detection. *ACM SIGKDD Explorations Newsletter*, 18(1): 1–14.

Zamini, M.; and Hasheminejad, S. M. H. 2019. A comprehensive survey of anomaly detection in banking, wireless sensor networks, social networks, and healthcare. *Intelligent Decision Technologies*, 13(2): 229–270.

Zhang, H.; and Davidson, I. 2021. Towards fair deep anomaly detection. In *Proc. FAccT*, 138–148. New York, NY: ACM.

Zhang, W.; and He, X. 2017. An anomaly detection method for medicare fraud detection. *2017 IEEE International Conference on Big Knowledge (ICBK)*, 309–314.