24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

A Framework for Determining the Fairness of Outlier Detection

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

Abstract. Outlier detection (OD) is a widely studied problem whose goal is to identify points from a data set that are considered anomalous. Among the methods used in AI and data science, OD is perhaps the most controversial as common applications such as credit card fraud, cyber-intrusion and terrorist activity all involve suggesting that someone is committing a serious crime. However, there is little work on fair outlier detection. We show how to determine if an outlier detection algorithm's output is fair with respect to *multiple* protected status variables (PSVs) by formulating various combinatorial problems which attempt to find an explanation (using the PSVs) that differentiates the outlier group from the normal group. We argue that if there is no solution for these explanation problems, then the output of an algorithm can be considered fair, and give a probabilistic interpretation of our work. Since we prove that the underlying combinatorial problems are computationally intractable (i.e., **NP**-hard), our approaches cannot be efficiently gamed/side-stepped.

1 Introduction & Motivation

Given a collection of points, the goal of outlier detection (OD) is to identify a small subset of points that are outliers, that is, unusual or anomalous or different. Hawkins [13] defined outliers as follows: "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". There are many methods for OD, and in our experimental section, we touch upon the main styles of outlier detection (namely, graph-based, distance-based, density-based, depth-based and derivation-based) [16]. OD is typically an unsupervised method and used extensively in many diverse fields such as security (intrusion detection), manufacturing (identifying flawed products) and even neuroscience (identifying individuals who have a certain neurological disorder). However, the division of points into outliers and normal points need not be generated by an algorithm: the notion of apriori-identified outliers by some external process (e.g., car lemon laws, product defect identification) has also been explored [22]. We call these *externally-defined* outliers. Regardless of whether the outliers are generated by algorithms, humans or by some other means, since OD is widely used in sensitive applications [5], it is important to develop techniques to assess the fairness of the outputs produced by OD methods.

The notion of fairness has recently received much attention in supervised learning (see e.g., [21]) with only recent work exploring unsupervised learning such as clustering (see e.g., [6, 11, 14, 20]). To our knowledge, there is no work on fairness in the context of OD. The fairness of an algorithm's output is typically measured with respect to a **single** protected status variable (PSV) such as gender, age, marital status, sexual-orientation, etc., which is **not** given to the algorithm. Measures of fairness can be divided at a high-level into two types [7]: (i) group-level fairness where we ensure that the PSV values are uniformly divided across both the normal points and the outlier points and (ii) individual-level fairness where we require two points which are very similar but with different protected status to be treated the same (i.e., both identified as normal or outlier).

Person	Gender	Status
1	`Male`	`Married`
2	`Female`	`Unmarried`
3	`Male`	`Married`
4	`Female`	`Unmarried`
5	`Male`	`Unmarried`
6	`Female`	`Married`
7	`Male`	`Unmarried`
8	`Female`	`Married`

Table 1. An illustrative collection of four points identified as outliers (Persons 1–4) and four points identified as normal (Persons 5–8).

A Motivating Example For Considering Multiple PSVs. Consider the simple example of an outlier detection method which outputs four people (Persons 1-4) as outliers and four people (Persons 5-8) as normal as shown in Table 1. Studying each PSV[3] *individually*, we see that the collection of outliers is group-wise fair: there are 50% `Females` and 50% `Males` and there are 50% `Unmarried` people and 50% `Married` people which is the same as the population averages and the normal group's averages. However, analyzing the two PSVs *together*, we see an indication of unfairness: all `Male` and `Married` individuals and all `Female` and `Unmarried` individuals are in the outlier group and none in the normal group. Our combinatorial optimization formulations attempt to find combinations of PSV values that are more probable in the outlier group than the population or normal group. How more probable they are is defined by the parameters set by the domain expert.

In this paper we explore a method to check whether an OD algorithm's **output** is fair. We take a combinatorial optimization approach to make the method general purpose and independent of the outlier detection algorithm. While presenting our *analytical* results, we limit our attention to binary PSVs. However, when conducting our experiments, we use one-hot encoding to convert non-binary PSVs to a

[1] Department of Computer Science, University of California - Davis, CA, USA. Email: davidson@cs.ucdavis.edu.
[2] Biocomplexity Institute & Initiative, University of Virginia, Charlottesville, VA, USA and Computer Science Department, University at Albany – SUNY, Albany, NY, USA. Email: ssravi0@gmail.com.

[3] Throughout the technical discussion in this paper, we assume for simplicity that PSVs such as `Gender`, `Martial Status`, etc. are binary. But in our experiments we show that our method can handle multi-state PSVs by using one-hot encoding.

binary encoding.

Core Idea and Role of the Human. With m PSVs, there are $2^m - 1$ combinations (nonempty subsets) of them. In our work, an explanation is comprised of a combination of PSVs. We define a computationally intractable (i.e., **NP**-hard) combinatorial problem which attempts to find explanations under different constraints on how many of the outlier and normal group are explained/covered. If the problem has a solution, then the OD algorithm's output can be regarded as unfair and our method gives the explanation why. If there is no solution, then the output can be deemed fair, given the constraints we imposed on the optimization problem. For example, we may say that the OD algorithm's output is unfair if the probability of a PSV combination (an explanation) occurring in the outlier group is greater than 50% or the population average. Most importantly, the use of a computationally intractable problem for assessing the fairness means that no OD algorithm can efficiently overcome/game the assessment method, under the common assumption that the complexity classes **P** and **NP** are different [12]. The human plays a pivotal role in our work, they set the parameters of our algorithms to determine what is "unfair" and interpret the explanation to determine if it is indeed a case of unfairness.

Organization. In Section 2 we formalize the notions of explanation and coverage used throughout our work. We then formulate our approach of determining whether an OD algorithm's output is fair as integer linear programs (ILPs) in Section 3 and how a human expert can use them in Section 4. In Section 5 we establish the complexity of our formulations to show that efficient OD methods cannot readily game or sidestep our approach (under a commonly used assumptions in complexity theory). Section 6 shows experimental results to complement our theoretical contributions. Section 7 concludes the paper.

2 Notation and Definitions

2.1 Cover and Anti-Cover

Let $\mathbb{O}$ and $\mathbb{N}$ denote the set of outlier and normal points so that $\mathbb{D} = \mathbb{O} \cup \mathbb{N}$ denotes the set of all data points. It is assumed that sets $\mathbb{N}$ and $\mathbb{O}$ are disjoint and produced by an OD algorithm. Let $\mathbb{P}$ denote the set of PSVs. Each PSV $p \in \mathbb{P}$ is assumed to take on a value from $\{0,1\}$. As mentioned earlier and as we do in our experiments, multi-state PSVs which can take on one of r values can be encoded as r binary PSVs. For any point $x \in \mathbb{D}$ and any PSV $p \in \mathbb{P}$, let $x(p)$ denote the value of p for x.

Definition 2.1 ***(Cover)***
*(a) Given a point $x \in \mathbb{D}$, a PSV $p \in \mathbb{P}$ **covers** x if $x(p) = 1$.*
*(b) A subset P' of $\mathbb{P}$ forms a **cover** for a set $S \subseteq \mathbb{D}$ of points if for each point $x \in S$, there is a PSV $p \in P'$ that covers x (i.e., $x(p) = 1$).*

Definition 2.2 ***(Anti-cover)***
*A subset P' of $\mathbb{P}$ forms an **anti-cover** for a set $S \subseteq \mathbb{D}$ of points if for each point $x \in S$ and every PSV $p \in P'$, $x(p) = 0$ (i.e., no PSV in P' covers any point in S).*

One can visualize a cover and anti-cover in a Venn diagram as shown in Figure 1. If Y is a region representing a set of points, then a cover for Y is a region which contains all of Y. Further, an anti-cover for Y is a region which has no overlap with Y.

We can relax the definition of cover via the notion of an f-cover which only requires that *at least* a fraction f of a set of points are covered. Similarly, a g-anti-cover is a relaxation of the notion of anti-cover; a g-anti-cover permits *at most* a fraction g of a set points to be covered.

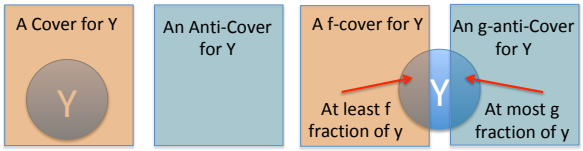

Figure 1. A Venn diagram view of the notion of cover, anti-cover, f-cover and g-anti-cover for a set of points Y.

2.2 A Vector-Based Notation

For convenience in formulating integer linear programs (ILPs) of our combinatorial problems, we introduce a vector based notation. All vectors are assumed to be *column* vectors. Recall that any OD algorithm effectively *partitions* the given data set $\mathbb{D}$ into two subsets, namely $\mathbb{O}$ and $\mathbb{N}$. Let $\mathbb{P} = \{p_1, p_2, \ldots, p_m\}$, where $m = |\mathbb{P}|$, denote the set of PSVs. For fairness considerations, each point in $x \in \mathbb{D}$ is described by a binary vector with m components, with the k^{th} component giving the value $x(p_k)$, $1 \leq k \leq m$. We use $\mathbb{O}_i$ to denote the binary vector corresponding to the i^{th} point in $\mathbb{O}$. Similarly, we use $\mathbb{N}_j$ to denote the binary vector corresponding to the j^{th} point in $\mathbb{N}$. Any subset P' of $\mathbb{P}$ can also be represented by a Boolean vector $\mathbf{x}$ with m components, where the k^{th} component is 1 if $p_k \in P'$ and 0 otherwise, $1 \leq k \leq m$. For a vector $\mathbf{x}$, we use $\mathbf{x}^T$ to denote its transpose.

Under this notation, if a subset P' of PSVs, represented by the vector $\mathbf{x}$, forms a cover for $\mathbb{O}$ and anti-cover for $\mathbb{N}$, then it can be seen that the following conditions hold:

$$\begin{array}{lcll} \mathbf{x}^T \mathbb{O}_i & \geq & 1 & \forall i,\ 1 \leq i \leq |\mathbb{O}| \quad \text{and} \\ \mathbf{x}^T \mathbb{N}_j & = & 0 & \forall j,\ 1 \leq j \leq |\mathbb{N}|. \end{array}$$

3 ILP Formulations and Probabilistic Interpretation

3.1 Overview

The previous section outlined constraints for a subset of PSVs to form a cover or an anti-cover. Here we use those constraints to construct several optimization problems with each being a test of fairness. In each case, the optimization problem tries to find an explanation (using the PSVs) that differentiates the outlier points from the normal points. If such an explanation exists, then the OD algorithm's output is unfair and our method produces the explanation why. If there is no explanation then the output is fair, given the constraints used.

High Level Description. We now outline the three optimization problems explored in this paper descending from most strict to least strict. *The most strict version, called **valid outlier description** (or **VOD**), is only for illustrative purposes and not used in our experimental results.* It can be seen that each problem is a variant of the other. The objectives of these three problems are shown diagrammatically in Figure 2. That figure shows a Venn diagram where the points are divided into two groups (normal and outliers) and the coverage of the explanation ($\mathbf{x}$) is denoted by a black dashed rectangle. This

enables one to easily understand the three objectives of our optimization problems:

- **VOD-Unfairness** (Left panel in Figure 2). Here we require finding an explanation to cover *only* the outlier points and *none* of the normal points; that is, it is an anti-cover for them. This is included for illustrative purposes.
- **α-VOD Unfairness** (Middle panel in Figure 2). Here we require finding an explanation to cover *all* the outlier points; however, such an explanation may cover *some* (at most α) of the normal points and hence is an $(\alpha/|\mathbb{N}|)$-anti-cover for them.
- **(α, β)-VOD-Unfairness** (Right panel in Figure 2). Here we require *simultaneously* finding a subset $\mathbb{O}'$ of the outliers obtained by removing at most β points from $\mathbb{O}$ and an α-VOD for the set $\mathbb{N}$ and $\mathbb{O}'$. In other words, the chosen descriptor must cover at least $|\mathbb{O}| - \beta$ points of $\mathbb{O}$ and at most α points of $\mathbb{N}$; in other words, it is a $(1 - \beta/|\mathbb{O}|)$-cover for $\mathbb{O}$ and an $(\alpha/|\mathbb{N}|)$-anti-cover for $\mathbb{N}$.

3.2 A Probabilistic Interpretation

We explore a probabilistic interpretation of our work when an explanation is found; that is, the OD algorithm's output is deemed unfair.

VOD. Suppose we apply the VOD problem formulation to a data set. For VOD-unfairness, when there is no solution to the problem, we can conclude that for *every* descriptor (i.e., subset of PSVs) that occurs in the outlier group, there is *at least one* point in the normal group for which the descriptor is true. Conversely, if the problem has a solution such as X = `female OR poor`, then it means everyone who is $\neg$`(female OR poor)` must be in the normal group (due to the anti-cover requirement for $\mathbb{N}$). Hence, the output is unfair to anyone who is either `female OR poor` because there exists an X such that $P(\text{Outlier}|X) = 1$ and $P(\text{Normal}|X) = 0$.

α-VOD. Suppose we apply the α-VOD problem formulation. Suppose (for simplicity) the same explanation (i.e., X = `female OR poor`) was found (because the dataset has changed) for α-VOD where $\alpha = |\mathbb{O}|$. Thus, the number of people satisfying the condition `female OR poor` in $\mathbb{O}$ is $|\mathbb{O}|$ and the number of people satisfying the same condition in $\mathbb{N}$ is some integer $\alpha' \leq \alpha = |\mathbb{O}|$. Hence, $P(\text{Outlier}|X) = |\mathbb{O}|/(|\mathbb{O}| + \alpha') \geq 0.5$ since $\alpha' \leq |\mathbb{O}|$. This makes the output of the OD algorithm unfair since there is a PSV combination that is more probable in the outlier group than the normal group.

(α, β)-VOD. A limitation of the previous two formulations is that they assume an explanation for *all* points in $\mathbb{O}$. This precludes identifying the situation where a small fraction of individuals with a rare PSV combination are discriminated against. The (α, β)-VOD formulation addresses this situation. This formulation searches for an explanation for at least $|\mathbb{O}| - \beta$ instances from $\mathbb{O}$ but also explains at most α instances from $\mathbb{N}$. Suppose we set $\alpha = |\mathbb{O}| - \beta$ and there is a solution X. Then the number of points covered in $\mathbb{O}$ is at least $|\mathbb{O}| - \beta$ and the number α' of points covered in $\mathbb{N}$ is at most $\alpha = |\mathbb{O}| - \beta$. Thus, $P(\text{Outlier}|X) \geq (|\mathbb{O}| - \beta)/[(|\mathbb{O}| - \beta) + (|\mathbb{O}| - \beta)] \geq 0.5$ if there is a solution. If there is no solution, then no subset of PSV values exist that are more probable in the outlier group than in the normal group.

3.3 An ILP for the VOD-Unfairness Detection Problem

Using the concepts in Section 2 we can construct an optimization problem to determine if a VOD exists. As mentioned earlier, our method can handle single and multiple variable unfairness using one-hot encoding. In our problem formulations, `argmax` finds the most specific combination whilst `argmin` finds the most general. For the remainder of this paper, we use $||\mathbf{x}||$ to denote the number of 1's in the vector $\mathbf{x}$.

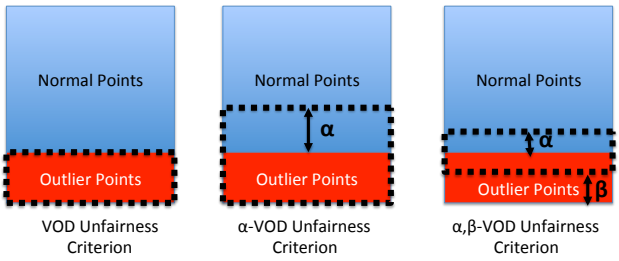

Figure 2. A Venn diagram view of the three optimization problems we address in this paper and formalize in Section 3. The explanation of the instances found is shown by the black dashed line.

Definition 3.1 ***The VOD-Unfairness Detection.***

$$\{argmin, argmax\}_{\boldsymbol{x}} \; ||\boldsymbol{x}|| \quad \text{such that}$$

$$\begin{aligned} \boldsymbol{x}^T \mathbb{O}_i &\geq 1 \quad \forall i \\ \boldsymbol{x}^T \mathbb{N}_j &= 0 \quad \forall j \end{aligned}$$

When there is a solution, the output of the above optimization problem is be a **subset** of PSVs (encoded in the vector $\mathbf{x}$) that can be used to differentiate outliers from normal points. If no solution exists, then the OD algorithm's output can be regarded as fair under the strict definition of fairness.

We show in Section 5 that the `argmax` version of the above problem can be solved efficiently while the `argmin` version is **NP**-hard.

3.4 An ILP for the α-VOD-Unfairness Detection Problem

Definition 3.1 is quite a strict definition of unfairness. If just one point in the normal group satisfies the description corresponding to $\mathbf{x}$ for the outlier points, the output would be deemed fair. Thus, it is useful to relax this condition by requiring that of the $|\mathbb{N}|$ normal points, at most α of them also be covered by $\mathbf{x}$, for some integer α.

Let $m = |\mathbb{P}|$ be the number of PSVs. To develop an ILP for this version, in addition to the {0,1}-valued variables in $\mathbf{x}$, we introduce $\gamma = |\mathbb{N}|$ additional {0,1}-valued variables, denoted by $y_1, y_2, \ldots, y_\gamma$, one corresponding to each point in $\mathbb{N}$. We will create constraints so that $y_j = 1$ if a chosen vector $\mathbf{x}$ covers the j^{th} point in $\mathbb{N}$; otherwise, $y_j = 0$, $1 \leq j \leq \gamma$. In this manner, $\sum_{j=1}^{\gamma} y_j$ will give us the number of points of $\mathbb{N}$ covered by the vector $\mathbf{x}$. The ILP is given below.

Definition 3.2 ***α-VOD-Unfairness Detection.***

$$\{argmin, argmax\}_{\boldsymbol{x}} \; ||\boldsymbol{x}|| \quad \text{such that}$$

$$\begin{aligned} \boldsymbol{x}^T \mathbb{O}_i &\geq 1 \quad \forall i \\ y_j &\leq \boldsymbol{x}^T \mathbb{N}_j \quad \forall j \\ m\, y_j &\geq \boldsymbol{x}^T \mathbb{N}_j \quad \forall j \\ \textstyle\sum_j y_j &\leq \alpha \end{aligned}$$

In the above ILP, all the variables in **x** and all the auxiliary variables $y_1, y_2, \ldots, y_\gamma$ take on values from {0,1}.

We now explain how the above ILP correctly models the α-VOD-Unfairness Detection Problem. The constraint on $\mathbb{O}$ ensures that each point of $\mathbb{O}$ is covered by the chosen vector **x**. The constraint $y_j \leq \mathbf{x}^T \mathbb{N}_j$ $(\forall j)$ ensures that when the right hand side of this constraint is 0, y_j must be set to 0. (In other words, if **x** does not cover the j^{th} point in $\mathbb{N}$, then $y_j = 0$.) Similarly, the constraint $my_j \geq \mathbf{x}^T \mathbb{N}_j$ $(\forall j)$ ensures that when the right hand side (RHS) is ≥ 1, y_j must be set to 1. (Thus, if **x** does cover the j^{th} point in $\mathbb{N}$, then $y_j = 1$.) It may help the reader to recall that the size of **x** is m; so, the maximum value of the RHS of this constraint is m. Hence, these constraints ensure that the value of y_j correctly indicates whether or not the corresponding point of $\mathbb{N}$ is covered by **x**. Finally, the constraint $\sum_j y_j \leq \alpha$ ensures that at most α normal points are covered, as required by the problem specification.

Setting the Parameter α. If $\alpha = |\mathbb{N}|$ and no solution is found, this can be interpreted to mean that any explanation that covers all $|\mathbb{O}|$ outliers also explains at least that many normal points, making the output fair. This naturally gives us a quantitative measure of fairness: the larger the value of α for which there is no solution for a given data set, the fairer is the output.

In Section 5, we will show that even the problem of determining whether there is a solution **x** that satisfies all the constraints of α-VOD-Unfairness Detection problem is **NP**-complete. Thus, the problem is computationally intractable even without any optimization objective for $||\mathbf{x}||$.

3.5 An ILP for the (α, β)-VOD-Unfairness Detection Problem

Our previous formulations require coverage of **all** outlier detection points. However, there maybe situations where we wish to just identify a very precise subset of outliers which are unfair. We formulate this problem as a combinatorial optimization problem that involves explaining at most α points in $\mathbb{N}$ as before (i.e., at most α points in $\mathbb{N}$ may be covered) but now require ignoring/not-explaining at most β points in $\mathbb{O}$. The ILP formulation for this problem, discussed below, is similar to that for the α-VOD-Unfairness Detection problem discussed in the previous subsection.

Recall that $m = ||\mathbb{P}|$ denotes the number of PSVs. To develop an ILP for this version, in addition to the {0,1}-valued variables in **x**, as above we introduce (i) $\gamma = |\mathbb{N}|$ additional {0,1}-valued variables, denoted by $y_1, y_2, \ldots, y_\gamma$, one corresponding to each point in $\mathbb{N}$, and (ii) $\tau = |\mathbb{O}|$ additional {0,1}-valued variables, denoted by $z_1, z_2, \ldots, z_\tau$, one corresponding to each point in $\mathbb{O}$. As done previously, we create the constraints so that $y_j = 1$ ($z_i = 1$) if a chosen vector **x** covers the j^{th} (i^{th}) point in $\mathbb{N}$ ($\mathbb{O}$); otherwise, $y_j = 0$, $1 \leq j \leq \gamma$ ($z_i = 0$, $1 \leq i \leq \tau$). In this manner, $\sum_{j=1}^{\gamma} y_j$ and $\sum_{i=1}^{\tau} z_i$ will give us the number of points of $\mathbb{N}$ and $\mathbb{O}$ covered by the vector **x** respectively. The ILP is given below.

Definition 3.3 (α, β)***-Unfairness Detection.***

$$\{argmin, argmax\}_{\boldsymbol{x}} \ ||\boldsymbol{x}|| \quad \text{such that}$$

$$\begin{array}{lcll} y_j & \leq & \boldsymbol{x}^T \mathbb{N}_j & \forall j \\ m\, y_j & \geq & \boldsymbol{x}^T \mathbb{N}_j & \forall j \\ z_i & \leq & \boldsymbol{x}^T \mathbb{O}_i & \forall i \\ m\, z_i & \geq & \boldsymbol{x}^T \mathbb{O}_i & \forall i \\ \sum_j y_j & \leq & \alpha & \\ \sum_i z_i & \geq & |\mathbb{O}| - \beta & \end{array}$$

In the above ILP, all the variables in **x** and all the auxiliary variables y_j $(1 \leq j \leq \gamma)$ and z_i $(1 \leq i \leq \tau)$ take on values from {0,1}. The constraints on $\sum_j y_j$ and $\sum_i z_i$ ensure that at most α normal points are covered and at most β outlier points are not covered. The rest of the argument to show that the above ILP correctly represents the (α, β) Unfairness Detection Problem is similar to the one given for the α-VOD-Unfairness Detection problem.

In Section 5, we will show a complexity result for this problem similar to that for the α-VOD-Unfairness Detection problem.

4 Using Our Work In Practice and the Role of the Domain Expert

We have presented in the prior sections the fundamentals of our approach. Here we discuss how a domain expert can use them in practice. We omit discussion of VOD as it is only an illustrative setting. We first overview how a user can set the parameters α and β and then how to use our method systemically.

Setting Parameters. Each domain will have different thresholds for identifying unfairness and these can naturally be modeled using the parameters α and β.

The α VOD approach is a method of detecting systemic (or global/wide-scale) bias in the identified outliers. This is because **all** instances in the outlier group must be covered by the explanation X (e.g., everyone flagged as an outlier satisfies the condition `unmarried` OR `female`. The parameter α determines how many people in the normal group are also covered by X. How we set α can determine what threshold we use to determine unfairness. Several possible examples are given below.

- Outlier/Normal ratio unfairness: Here if a pattern is more probable in the outlier group than in the normal group, then it is deemed unfair. For example, suppose we set $\alpha = |\mathbb{O}|$ (as describe earlier). Then if an explanation X is found, we have then $P(\text{Outlier}|X) > 0.5$.
- Outlier/Population ratio unfairness: Here if a pattern is more probable in the outlier group than in the entire population, then it is deemed unfair. Suppose we set $\alpha = (|\mathbb{O}|/|\mathbb{D}|) * |\mathbb{N}|$. Then if an explanation X is found, we have then $P(\text{Outlier}|X) > P(\text{Normal}|X)$. That is, the probability of finding the X in the outlier group is greater than finding X in the normal group.

The (α, β)-VOD approach is useful for finding specific (or local, fine-grained) bias. It allows a small subset of the outlier group (no less than $|\mathbb{O}| - \beta$) to be explained/identified using X. This approach has several benefits.

- It can model the classic **disparate impact unfairness** [3]. The 80% rule of disparate impact requires that if for example 4 men are identified as outliers then no more than 5 woman should be identified as outliers. To enforce this, we can simply require that $\beta = \frac{4}{9} * |\mathbb{O}|$. If a solution X is found, then the ratio of X found in $\mathbb{O}$ must be $\geq \frac{4}{9}/\frac{5}{9} \geq 0.8$.
- It can model Outlier/Population ratio unfairness described above but now for a subset of the outliers. Suppose we set $\alpha = ((|\mathbb{O}| - \beta)/|\mathbb{D}|) * |\mathbb{N}|$. Then, if an explanation X is found, we have $P(\text{Outlier}|X) > P(\text{Normal}|X)$.

Human in the Loop Extensions. The identification of unfairness in an OD algorithm's output can be used as a filter for further human examination. Here we describe a Test→Exclude→Test Loop that allows our work and results to be used together. Let there be

m binary PSVs. There are $2^m - 1$ possible (nonempty) subsets (i.e., disjunctions) of the PSVs. All our methods will discover if any of these $2^m - 1$ disjunctions is unfair given various definitions of fairness (whose probabilistic interpretation was discussed above). The domain expert can then examine the explanation (X) found to determine if it is a true example of unfairness. If it is, then the loop is terminated and the output of the OD algorithms is deemed unfair. If not, then X can be excluded from future searches with a simple constraint, namely $X \neq X_{Old}$.

Having a human verify that the discovered combinatorial unfairness is an actual case of unfairness is the only "iron-clad" way to determine fairness in this context.

5 Complexity of Unfairness Detection

5.1 Overview

This section outlines the complexity of the three unfairness detection problems defined by ILPs in Section 3. The results presented in this section include the following.

1. We show that the `argmax` version of the strict VOD-unfairness problem can be solved efficiently, while the `argmin` version of the same problem (formulated suitably as a decision problem) is **NP**-complete.
2. For the α-VOD-Unfairness and (α, β)-VOD-Unfairness problems, we show that even determining whether a solution exists is **NP**-complete. Thus, these problems are computationally intractable even without any optimization requirement on the number of PSVs chosen in the explanation (i.e., descriptor).

To prove complexity results, we need to reformulate each of the problems defined in the previous section as a decision problem. These reformulations are presented in the ensuing subsections. It is straightforward to verify that these decision versions indeed correctly represent the corresponding unfairness problems.

5.2 Results for VOD-Unfairness Detection

We begin with the basic decision problem corresponding to the VOD-Unfairness Detection problem.

(a) Valid Outlier Descriptor Existence (VODE)

Given: Sets $\mathbb{O}$ and $\mathbb{N}$ of outlier and normal points and the set $\mathbb{P}$ of PSVs.

Question: Is there a subset $P' \subseteq \mathbb{P}$ such that P' is a valid outlier descriptor for $\mathbb{O}$ and $\mathbb{N}$ (i.e., P' covers $\mathbb{O}$ and is an anti-cover for $\mathbb{N}$)?

Algorithm 1: An algorithm for VODE

```
1 Set P_1 = ℙ.
2 for each p ∈ ℙ do
3 |   If p covers some point in ℕ, then remove p from P_1.
4 end
5 if P_1 is a cover for 𝕆 then
6 |   output P_1 as the solution.
7 else
8 |   output "No solution".
9 end
```

Theorem 1 *The VODE problem can be solved efficiently. If a valid outlier descriptor exists, then such a descriptor of* ***maximum*** *size can also be found efficiently.*

Proof: A simple algorithm for VODE is given in Algorithm 1. The idea is to find a subset P_1 of $\mathbb{P}$ such that P_1 contains only those PSVs which *do not* cover any point in $\mathbb{N}$. The algorithm then checks whether P_1 forms a cover for $\mathbb{O}$. If so, the algorithm outputs P_1 as the solution; otherwise, the algorithm outputs the message "No solution". We now prove the correctness of the algorithm. We also show that if P_1 is a solution, then it has the largest cardinality among all the solutions.

Part 1: Suppose P_1 is a cover for $\mathbb{O}$. Since P_1 does not contain any PSV which covers a point in $\mathbb{N}$, P_1 is also an anti-cover for $\mathbb{N}$. Hence, P_1 is a valid outlier descriptor.

Part 2: Suppose P_1 does not cover all the points in $\mathbb{O}$. We will prove by contradiction that there is no VOD for the given VODE problem instance. So, assume that Q is a solution and consider any variable $q \in Q$. Since Q is an anti-cover for $\mathbb{N}$, q cannot cover any point in $\mathbb{N}$. Thus, the iterative procedure that constructs P_1 would not have eliminated q. In other words, $q \in P_1$ and thus, $Q \subseteq P_1$. Now, since P_1 does not cover all the points in $\mathbb{O}$ and $Q \subseteq P_1$, Q also cannot cover all the points in $\mathbb{O}$. This contradicts the assumption that Q is a solution. Thus, if P_1 cannot cover all the points in $\mathbb{O}$, there is no solution.

We now observe that if the set P_1 constructed above is a solution to the VODE problem instance, then no other solution can have a larger cardinality. This follows from the argument presented in Part 2 above where it is shown that if Q is another solution, then $Q \subseteq P_1$.

We will present a simple running time analysis to show that Algorithm 1 runs in polynomial time. Assume that $\mathbb{N}$ is represented by a $|\mathbb{N}| \times |\mathbb{P}|$ matrix M_N such that the entry $M_N[i, j]$ gives the value of the PSV $p_j \in \mathbb{P}$ for the i^{th} point in $\mathbb{N}$. In a similar fashion, assume that $\mathbb{O}$ is represented by a $|\mathbb{O}| \times |\mathbb{P}|$ matrix M_O. Each iteration of the loop in Step 2 can be implemented to run in $O(|\mathbb{N}|)$ time using the matrix M_N. Thus, Step 2 runs in $O(|\mathbb{N}|\,|\mathbb{P}|)$ time. To check whether P_1 covers $\mathbb{O}$ (Step 5) we again need to check that each point in $\mathbb{O}$ is covered by some PSV in P'. This can be done in $O(|\mathbb{O}|\,|P'|) = O(|\mathbb{O}|\,|\mathbb{P}|)$ time using the matrix M_O. So, the overall running time is $O(|\mathbb{P}|\,(|\mathbb{N}| + |\mathbb{O}|))$. ■

As shown above, one can efficiently check whether there is a valid outlier descriptor, and if so, find one of *maximum* size. It is of interest to investigate whether one can find such a descriptor of *minimum* size (i.e., a descriptor with the smallest number of PSVs). We now show that this minimization version is **NP**-complete. We begin with a formulation the corresponding decision problem.

(b) Minimum Valid Outlier Descriptor (MVOD)

Given: Sets $\mathbb{O}$ and $\mathbb{N}$ of outlier and normal points, the set $\mathbb{P}$ of PSVs and an integer $k \leq |\mathbb{P}|$.

Question: Is there a subset $P' \subseteq \mathbb{P}$ such that $|P'| \leq k$ and P' is a valid outlier descriptor for $\mathbb{O}$ and $\mathbb{N}$?

Theorem 2 *The MVOD problem is* ***NP****-complete.*

See proof in technical report [10].

5.3 Complexity of α-VOD-Unfairness

Here, we consider the α-VOD-Unfairness problem and show that the problem of determining whether a descriptor exists (with no constraint on the number of PSVs in the descriptor) is **NP**-complete. It

follows that both `argmin` and `argmax` versions of the problem (as formulated in Section 3) are **NP**-hard. The decision version of the problem is as follows.

α-VOD-Unfairness (α-VOD)

Given: Sets $\mathbb{O}$ and $\mathbb{N}$ of outlier and normal points, the set $\mathbb{P}$ of PSVs and an integer $\alpha \leq |\mathbb{N}|$.

Question: Is there a subset $P' \subseteq \mathbb{P}$ of PSVs such that P' covers $\mathbb{O}$ and at most α points in $\mathbb{N}$?

Note that this decision problem is a generalization of the VODE problem considered earlier; if we set $\alpha = 0$, we obtain exactly the VODE problem. However, unlike the VODE problem, this problem is **NP**-complete as shown below.

Theorem 3 *The α-VOD problem is **NP**-complete.*

Proof: It is easy to see that α-VOD is in **NP**. We prove the **NP**-hardness is through a reduction from the **Minimum Set Cover** (MSC) problem [12]: given a universal set $U = \{u_1, u_2, \ldots, u_n\}$, a collection $C = \{C_1, C_2, \ldots, C_m\}$ of subsets, where $C_j \subseteq U$, $1 \leq j \leq m$, and an integer $k \leq m$, is there a subcollection $C' \subseteq C$ with $|C'| \leq k$ such that the union of the sets in C' is equal to U? The reduction is as follows.

1. The set $\mathbb{O} = \{o_1, o_2, \ldots, o_n\}$ of outliers is in one-to-one correspondence with the set $U = \{u_1, u_2, \ldots, u_n\}$.
2. The set $\mathbb{N} = \{\nu_1, \nu_2, \ldots, \nu_\gamma\}$ has $\gamma = 1 + \max\{n^2, 2m\}$ normal points. Of these, the subset $\mathbb{N}' = \{\nu_1, \nu_2, \ldots, \nu_m\}$ consisting of the first m elements is in one-to-one correspondence with the collection C = $\{C_1, C_2, \ldots, C_m\}$. (The choice of γ ensures that $|\mathbb{N}| > |\mathbb{O}|$.)
3. The set of PSVs $\mathbb{P} = \{p_1, p_2, \ldots, p_m, p_{m+1}\}$ has $m+1$ elements. Of these, the subset $P_1 = \{p_1, p_2, \ldots, p_m\}$ consisting of the first m PSVs is in one-to-one correspondence with the collection C = $\{C_1, C_2, \ldots, C_m\}$.
4. Suppose the element u_i, $1 \leq i \leq n$, appears in subsets C_{i_1}, C_{i_2}, $\ldots$, C_{i_r} for some $r \geq 1$. Then, for the outlier point o_i, the PSVs $p_{i_1}, p_{i_2}, \ldots, p_{i_r}$ have the value 1 and the remaining PSVs have value 0. For each normal point $\nu_j \in \mathbb{N}'$, $1 \leq j \leq m$, the PSV p_j has the value 1 and the remaining PSVs (including p_{m+1}) have the value 0. For each normal point $\nu_j \in \mathbb{N}$, $m+1 \leq j \leq \gamma$, the PSV p_{m+1} has the value 1 and all the remaining PSVs have the value 0.
5. The upper bound α on the number of normal points that can be covered is set to k (from the MSC problem).

This completes the polynomial time reduction. We now prove that there is a solution to the α-VOD problem iff there is a solution to the MSC problem.

Suppose $C' = \{C_{j_1}, C_{j_2}, \ldots, C_{j_\ell}\}$, where $\ell \leq k$, is a solution to the MSC problem. We claim that the subset $P' = \{p_{j_1}, p_{j_2}, \ldots, p_{j_\ell}\}$ is a solution to the α-VOD problem. We first show that P' forms a cover for $\mathbb{O}$. To see this, consider any element $o_i \in \mathbb{O}$. Since C' is a solution to MSC, the element $u_i \in U$ corresponding to o_i appears in some set, say $C_{j_y} \in C'$. By our construction of P', the variable p_{j_y} is in P' and o_i is covered by p_{j_y}. Thus, P' forms a cover for $\mathbb{O}$. We now argue that P' covers at most $\ell \leq k = \alpha$ elements in $\mathbb{N}$. To see this, notice that in the subset $\mathbb{N}' = \{\nu_1, \nu_2, \ldots, \nu_m\}$, each point ν_j can be covered by only one PSV, namely p_j, $1 \leq j \leq m$. Since $|P'| = \ell$, only ℓ points in $\mathbb{N}'$ can be covered by P'. Since P' does not contain p_{m+1} and each point in $\mathbb{N} - \mathbb{N}'$ can only be covered by p_{m+1}, we see that P' does not cover any point of $\mathbb{N} - \mathbb{N}'$. Thus, P' covers only $\ell \leq \alpha$ points of $\mathbb{N}$. In other words, P' is a solution to the α-VOD problem.

Suppose $P' = \{p_{j_1}, p_{j_2}, \ldots, p_{j_\ell}\}$ is a solution to the α-VOD problem. We first show by contradiction that $p_{m+1} \notin P'$. To see this, suppose $p_{m+1} \in P'$. Then, all the points in $\mathbb{N} - \mathbb{N}'$ would be covered by P'. Since $|\mathbb{N} - \mathbb{N}'| \geq m + 1 > k = \alpha$, P' would cover more than α points of $\mathbb{N}$; in other words, P' cannot be a valid solution to α-VOD problem. This is a contradiction and we conclude that $p_{m+1} \notin P'$. We also claim that $|P'| = \ell \leq \alpha = k$. To see this, if $|P'| > \alpha + 1$, then since each PSV in P' covers one normal point in $\mathbb{N}'$, the number of normal points covered by P' would exceed α. Thus, $P' \subseteq \{p_1, p_2, \ldots, p_m\}$ and $|P'| \leq \alpha = k$. We now claim that the subcollection $C' = \{C_{j_1}, C_{j_2}, \ldots, C_{j_\ell}\}$ is a solution to the MSC problem. Since $\ell \leq \alpha = k$, the constraint on $|C'|$ is satisfied. To see that C' forms a solution to MSC, consider any element $u_i \in U$. Since P' is a solution to α-VOD, there is a variable, say $p_{j_y} \in P'$, that covers o_i, the point corresponding to $u_i \in U$. By our construction of C', the element u_i is covered by the set $C_{j_y} \in C'$. Thus, C' is a solution to the MSC problem, and this completes our proof. ■

5.4 Complexity of (α, β)-VOD-Unfairness

Recall that the goal of (α, β)-VOD-Unfairness is to determine whether there is a subset of PSVs that can cover at least $|\mathbb{O}| - \beta$ points of $\mathbb{O}$ while covering at most α points of $\mathbb{N}$. A formal statement of this decision problem is as follows.

(α, β)-VOD-Unfairness ((α, β)-VOD)

Given: Sets $\mathbb{O}$ and $\mathbb{N}$ of outlier and normal points, the set $\mathbb{P}$ of PSVs and positive integers α and β.

Question: Is there a subset $P' \subseteq \mathbb{P}$ such that P' covers at least $|\mathbb{O}| - \beta$ points in $\mathbb{O}$ and at most α points in $\mathbb{N}$?

Theorem 4 *The (α, β)-VOD-Unfairness problem is **NP**-complete.*

See technical report [10] for proof.

6 Experimental Results

Our experiments attempt to address several questions that complement our theoretical results in the previous section.

- How fair are existing OD algorithms' outputs according to α-VOD Definition 3.2 and (α, β)-VOD (Definition 3.3)? (Tables 3, 4)
- Can existing rule discovery methods be used to detect fairness (Table 6)? This forms a series of baselines for comparison albeit for methods not designed for the OD output fairness evaluation.
- What is the typical runtime of our algorithm on a variety of data sets where the outliers are already given? (Table 5).

Fairness of Existing an OD Algorithm's Output. We begin our experiments using a classic dataset used in many fairness papers (see e.g., [6]), namely Census. This data set consists of census information along continuous dimensions such as `age`, `wage`, `hours worked`, etc. and contains many PSVs. We experiment with the following PSVs: `Education`, `Married-status`, `Relationship`, `Race`, `Sex`. For non-binary PSVs, we use a one-hot encoding; the encoded values are shown in Table 2. Thus, the vector **x** that we solve for is not simply of length equal to the number of PSVs.

PSV	Possible Values
Education	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
Marital-status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
Relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
Race	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Sex	Female, Male

Table 2. The PSVs Used in our Experiments on the Census Data. In total, the length of the vector **x** is 36.

We then apply several standard outlier detection methods from the classic styles of outlier detection [16] (which are given in parentheses): i) Oddball [1] (graph-based), ii) $DB(\epsilon, \pi)$ [15] (distance-based), iii) LOF [4] (density-based), iv) ISODepth [19] (depth-based), v) Model [2] (deviation-based). We then apply our two tests of unfairness as per Definitions 3.1 and 3.2 with results reported in Table 3. Unsurprisingly, all methods were found to be fair according to strict illustrative test in Definition 3.1; this test is omitted from future experiments.

For Definition 3.2 we report the smallest value of α that yield no feasible solution and hence makes the OD algorithm output fair. This value of the smallest α to obtain fairness can be interpreted as a measure of how fair is the algorithm's outputs; the *larger* the value, the more fair is the method.

OD Algorithm	VOD Unfair? (Definition 3.1)	Smallest α Ensuring α-VOD Fairness (Definition 3.2)
Graph-Based [1]	10:No, 0:Yes	0.63
Distance-Based [15]	10:No, 0:Yes	0.41
Density-Based [4]	10:No, 0:Yes	0.53
Depth-Based [19]	10:No, 0:Yes	0.61
Deviation-Based [2]	10:No, 0:Yes	0.54

Table 3. The results of our two tests of fairness. Results are averaged over 10 trials as some algorithms are randomly seeded. Note the α value is reported as a fraction of total number of outliers identified by the method. This allows direct comparison as each OD method returns a different number of outliers. The larger the number the fairer is the output.

We then applied our fine grained approach (Definition 3.3) that ignores up to β outlier instances until no α-VOD solution is found. Here we set α so as to require as many normal points as there are outlier points to be covered by the explanation. Results are shown in Table 4. Again we search for the smallest β value to obtain fairness and here the **smaller** the value of β the better (as it means we have to ignore fewer outliers). Overall we found that OD methods that construct data structures (i.e., graph-based and depth-based) covered more normal points (α) and ignored fewer outlier points (β).

Tests On More Datasets and Scalability. Previously we measured the fairness of standard outlier detection algorithms. Here we report further results and test the scalability of our method on standard outlier detection data sets. For a number of outlier data sets where the outliers are *already given* (see `http://odds.cs.stonybrook.edu/`), we applied our method using only the discrete/categorical variables in those data sets as the PSVs. Note that these variables would not always be considered PSVs but the main aim here is to test scalability. We see that our approach overall is quite efficient but becomes less efficient as the number of outliers grows. This is as we need to find a cover for all outlier points. Results are shown in Table 5. For all the data sets, we find that our α-VOD test of unfairness identify the results as being unfair.

OD Algorithm	Smallest β (Definition 3.3) for $\alpha = \lvert\mathbb{O}\rvert$ fairness
Graph-Based	0.31
Distance-Based	0.53
Density-Based	0.41
Depth-Based	0.33
Deviation-Based	0.39

Table 4. The results of our fine grained approach to ensure fairness as defined in Definition 3.3 and averaged over 10 runs. For all experiments α is set to equal the number of outliers the method generated. Note the β value is reported as a fraction of total number of outliers identified by the method. The smaller the value of β, the fairer is the algorithm.

Data Set: $\lvert\mathbb{N}\rvert/\lvert\mathbb{O}\rvert$	Run Time (α-VOD Unfair?)
Lympho 142/6	0.5 (Yes)
Glass 205/9	0.3 (Yes)
Thyroid 3670/93	18.8 (Yes)
Satimage-2 5732/71 (No)	32.7 (Yes)
Pima 500/268	4.2 (Yes)
Shuttle 45586/3511	195.7 (Yes)
Http (KDDCUP99) 56226/2211	221.3 (Yes)
Smtp (KDDCUP99) 95126/30	1.6 (Yes)

Table 5. The data set and number of normal ($\mathbb{N}$) and outlier ($\mathbb{O}$) points. The run time in seconds using Gurobi on a 4 CPU Xeon Machine for our two measures of fairness. We report in parentheses if a solution is found (i.e., the output is unfair). For α-VOD we set $\alpha = |\mathbb{O}|$.

Baseline Comparison Results. Creating baselines for comparison is challenging since to our knowledge, our method is the first one for assessing fairness for outlier detection. We create baseline methods from rule generation methods, which attempt to find a set of rules that differentiate between the two classes, namely Normal and Outlier. The reasoning here is similar to our own method but without the guarantees that our computational intractability results give, namely that if a non-null model exists that can differentiate the outlier points from the normal points using the PSVs, then the OD algorithm's output can be regarded as unfair. Results are shown in Table 6. All rule generation methods are in the Weka package: J48 (equivalent of C4.5 [18]), Ripper [9], Bayesian Rule Induction - cn2 (BRI) [8]. We find that these methods are remarkably consistent with each other and predict fairness (only the null model was found) for most data sets where as our methods indicate that they are unfair. For many large data sets, the output is also said to be fair by these methods (Thyroid, Satimage, Pima, Shuttle, Http (KDDCUP99)). This may seem counter-intuitive, but it is important to realize we limit ourselves to only the categorical variables (the PSVs) when building these models.

The failure of these methods is not unexpected as their aim is to generate a predictive model to maximize accuracy and not an explanation. Hence for data sets with a small number of outlier points

and/or small number of PSVs, they are more likely to predict the null model (i.e., always predict the normal class) as it has a very large accuracy.

Data Set: $\mathbb{N}/\mathbb{O}$	**J48**	**Ripper**	**BRI**
Lympho 142/6	No	No	No
Glass 205/9	No	No	No
Thyroid 3670/93	No	Yes	No
Satimage-2 5732/71	No	No	No
Pima 500/268	Yes	No	No
Shuttle 45586/3511	No	Yes	No
Http (KDDCUP99) 56226/2211	No	Yes	No
Smtp (KDDCUP99) 95126/30	No	No	No

Table 6. Baseline tests of *unfairness* by determining if rule-based methods can differentiate $\mathbb{N}$ and $\mathbb{O}$ using the PSVs with a non null-model. The data set and number of normal ($|\mathbb{N}|$) and outlier ($|\mathbb{O}|$) points are given. Compare with our results in Table 5.

7 Conclusions, Limitations & Future Work

We propose novel tests of fairness for the output of outlier detection algorithms based on certain combinatorial optimization problems. These problems attempt to find a shortest explanation (using protected status variables) that differentiates the outlier class from the normal class. If such an explanation exists then the output is deemed unfair. Our tests have the benefit of having user tunable parameters that can encode the user's tolerance to fairness. Our α-VOD approach can find global unfairness by identifying a subset of PSVs (X) that all individuals in the outlier group have and that at most α people in the normal group have. If α is set to be the number of outliers then the criteria for unfairness is then simply $P(Outlier|X) > 0.5$. Our (α, β)-VOD allows finding local (finer-grained) unfairness by identifying a subset of outlier points that exhibit α-VOD unfairness. Our empirical results show that not surprisingly the output of five classic outlier detection methods are unfair (see Table 5) whilst baseline rule-based methods are easily misled into concluding that their outputs are fair, especially when the number of outliers is small and the number of PSVs is small (see Table 6). This is not surprising as these baselines were created for predictive purposes.

We now point out some limitations of our work. As mentioned earlier, due to the computational intractability of the underlying problems, the approach cannot be easily gamed/side-stepped. However, our computational intractability results are most useful when the number of PSVs is not small. (Formally, the unfairness detection problems defined in Section 3 are *fixed parameter tractable* [17] with respect to the parameter $|\mathbb{P}|$, that is, the number of PSVs.) If $|\mathbb{P}| = m$ is small, one can try all the 2^m subsets of $\mathbb{P}$ to check if any of them is a solution. (Even for $m = 20$, the number of subsets is only about a million.) Likewise, our intractability results are not as useful when the number of outliers is very small. For example, if $|\mathbb{O}| \leq c \log |\mathbb{N}|$ for some constant $c > 0$, the combinatorial problems formulated in the paper can be solved efficiently. This is because in such a case, one can find a subset P' of PSVs, where $|P'| \leq |\mathbb{O}|$, to cover all the outliers. Thus, the number of subsets of P' to be tried is $2^{c \log |\mathbb{N}|} = \mathbb{N}^c$, which is a polynomial since c is a constant.

In this paper we limited ourselves to studying traditional OD methods that just identify the outliers. We leave to future work the study of other styles of OD detection such as group, collective and contextual discussed in [5].

Acknowledgments: We thank the ECAI 2020 reviewers for providing helpful comments. This work was supported in part by NSF Grants IIS-1908530 and IIS-1910306 entitled "Explaining Unsupervised Learning: Combinatorial Optimization Formulations, Methods and Applications".

REFERENCES

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos, 'Oddball: Spotting anomalies in weighted graphs', in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 410–421. Springer, (2010).
[2] Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan, 'A linear method for deviation detection in large databases', in *Proc. ACM KDD*, pp. 164–169, (1996).
[3] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner, 'Scalable fair clustering', *To Appear in ICML*, (2019).
[4] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander, 'Lof: identifying density-based local outliers', in *SIGMOD Record*, pp. 93–104. ACM, (2000).
[5] V. Chandola, A. Banerjee, and V. Kumar, 'Anomaly detection: A survey', *ACM computing surveys (CSUR)*, **41**(3), 15:1–15:58, (2009).
[6] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii, 'Fair clustering through fairlets', in *Proc. NeurIPS*, pp. 5036–5044, (2017).
[7] A. Chouldechova and A. Roth. The frontiers of fairness in machine learning. ArXiv: 1810.08810v1, 2018.
[8] Peter Clark and Tim Niblett, 'The cn2 induction algorithm', *Machine learning*, **3**(4), 261–283, (1989).
[9] William W Cohen, 'Fast effective rule induction', in *Machine learning proceedings 1995*, 115–123, Elsevier, (1995).
[10] Ian Davidson and S. S. Ravi, 'Towards determining the fairness of outlier detection', Technical report, Univeristy of California - Davis, Davis, CA, USA, (2019). `www.cs.ucdavis.edu/~davidson/Publications/TR.pdf`.
[11] Ian Davidson and S. S. Ravi, 'Making existing clusterings fairer: Algorithms, complexity results and insights.', in *Proc. AAAI*, p. To Appear, (2020).
[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman & Co., San Francisco, 1979.
[13] Douglas M Hawkins, *Identification of outliers*, volume 11, Springer, New York, NY, 1980.
[14] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern, 'Fair k-center clustering for data summarization', in *Proc. ICML*, pp. 3448–3457, (2019).
[15] Edwin M Knorr, Raymond T Ng, and Vladimir Tucakov, 'Distance-based outliers: algorithms and applications', *The VLDB JournalThe International Journal on Very Large Data Bases*, **8**(3-4), 237–253, (2000).
[16] H-P. Kriegel, P. Kröger, and A. Zimek. Outlier detection techniques. Tutorial at SDM 2010, 2010. Slides available from: `https://archive.siam.org/meetings/sdm10/tutorial3.pdf`.
[17] R. Neidermeier, *Invitation to fixed parameter algorithms*, Oxford University Press, New York, NY, 2006.
[18] J Ross Quinlan, *C4. 5: programs for machine learning*, Elsevier, 2014.
[19] I Ruts and PJ Rousseeuw, 'Isodepth: A program for depth contours', in *COMPSTAT*, pp. 441–446. Springer, (1996).
[20] M. Schmidt, C. Schwiegelshohn, and C. Sholer, 'Fair coresets and streaming algorithms for fair k-means clustering', *CoRR*, **abs/1812.10854v1**, (2018).
[21] Binh Luong Thanh, Salvatore Ruggieri, and Franco Turini, 'k-NN as an implementation of situation testing for discrimination discovery and prevention', in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pp. 502–510, (2011).
[22] Cui Zhu, Hiroyuki Kitagawa, Spiros Papadimitriou, and Christos Faloutsos, 'Outlier detection by example', *Journal of Intelligent Information Systems*, **36**(2), 217–247, (2011).