Fairness in Clustering and Outlier Detection as Constraint Satisfaction

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

[1]Department of Computer Science, University of California Davis, Davis, CA, USA. [2]Biocomplexity Institute and Initiative, University of Virginia, Charlottesville, VA 22904 and Department of Computer Science, University at Albany – SUNY, Albany, NY 12222, USA. Email: indavidson@ucdavis.edu, ssravi0@gmail.com

Abstract

The topic of fairness has attracted a lot of attention in the machine learning literature (Barocas, Hardt, and Narayanan 2017) with the typical group-level fairness explored being disparate impact. Our AAAI 2020 paper and followup (Davidson and Ravi 2020b; Davidson et al. 2022) pointed out how disparate impact can be encoded as a cardinality constraint and that satisfying this constraint for a single protected status is efficiently solvable. We also showed how to use this constraint in an optimization setting to minimally modify a k-block set partition to make it fairer. Later work (Davidson and Ravi 2020a, 2022) studied auditing the output of outlier detection and clustering algorithms by searching for a under-representation (count) of a protected status combination in one group and an over-representation in another. We formulated integer linear programs (ILPs) for these problems since the underlying feasibility problems are **NP**-hard. We briefly summarize these papers and present some new directions for fairness we hope the CP community can help with.

1 Introduction

The AI community has made a lot of progress towards making algorithms fairer. Fairness has been studied in the context of many major machine learning (ML) tasks such as clustering, classification, ranking, embedding and anomaly detection. Fairness can be broadly broken down into two categories: individual-level fairness and group-level fairness. Our work focuses on the latter, and in particular, on a legal definition of fairness known as disparate impact. Generally speaking, disparate impact encodes the notion that the fraction of protected status individuals chosen by an algorithm (or associated with an action) must be approximately equal to the fraction of the protected status individuals in the general population. For example, if the general population contains 50% females, then approximately 50% of individuals selected for a job interview must be female.

We pointed out (Davidson and Ravi 2020b; Davidson et al. 2022) that disparate impact can be easily encoded as a cardinality constraint by counting the number of protected status individuals in a group and comparing it to a constant (the fraction of individuals in the population with the protected status). Our first work showed that satisfying disparate impact for a single protected status can be solved efficiently. This was done by showing that the underlying constraint matrix is totally unimodular. We also showed that satisfying a form of individual level fairness (Davidson and Ravi 2007, 2020b) is also efficiently solvable. However, while satisfying either individual-level fairness or group level fairness is efficiently solvable satisfying both individual- and group-level constraints is computationally intractable.

The above results are for a single protected status variable. Our later work (Davidson and Ravi 2020a, 2022) explored auditing the output of an algorithm that produces a k-block partition for fairness. This is done by searching for a protected status combination that is over-represented in one block and under-represented in another block. We formulated this as an ILP and showed that a variety of different counting constraints can be used to encode different forms of disparate impact-like fairness.

Next, we briefly summarize these papers and discuss future work in the area of fairness as constraint satisfaction we hope the CP community can contribute to.

2 Modifying a Clustering to Enhance Fairness

Many different clustering algorithms, with a variety of settings, formulations and followings by end user communities, are available (see e.g., (Xu and Wunsch 2005)). It is unlikely that fair versions of all these algorithms or future clustering algorithms will be developed. So, it is useful to study the setting where one already has a good clustering Π and the goal is to modify Π to improve its fairness.

The fairness measure considered in our work uses is based on **protected status variables** (PSVs). Our focus has been on the simplest but most common type of PSVs, namely binary variables such as gender. To provide a formal definition of our fairness measure, we introduce some definitions. Consider a data set D where there is one PSV x. The data items in D for which the variable x has value 1 will be referred to as **special** items. Suppose D, with N_x special items, has been partitioned into k clusters, denoted by C_1, C_2, . . ., C_k. Then the number of special data items per cluster could be set to be approximately N_x/k, to effectively balance the protected status instances uniformly across all clusters. This gives rise to one definition of fairness:

Definition 2.1. *Let D be a dataset where each data item has a single binary protected attribute x. Let N_x denote the number of special data items in D. A partition of D into $k \geq 2$ clusters is* ***strongly fair with respect to*** *x if in each cluster, the number of special items is either $\lfloor N_x/k \rfloor$ or $\lceil N_x/k \rceil$.*

Relaxed versions of this definition have also been considered in (Davidson and Ravi 2020b). In particular, to enforce fairness based on a common disparate impact criterion, one would require each cluster to have at least $0.8 \times N_x/k$ and at most $1.2 \times N_x/k$ special items.

The **Minimum Cluster Modification for Fairness** (MCMF) problem is the following: given an existing clustering (defined through a $k \times n$ allocation matrix Z, where k is the number of clusters and n is the number of instances), find a minimal modification that makes the clustering fairer with respect to a single PSV. A general linear function that can represent a variety of minimization objectives (e.g., the number of instances moved, increase distortion) is presented in (Davidson et al. 2022). Here, we will focus on the necessary constraints to achieve fairness.

(a) Constraints to achieve strong fairness: The constraints serve a two-fold purpose: to balance the protected instances across clusters whilst also restricting Z to be a legal cluster allocation matrix. For concreteness, we use the encoding where indicator vectors are stacked column-wise; that is $z_{i,j} = 1$ iff instance j is assigned to cluster i. We encode protected status as a vector P of length n with an entry of 1 if the instance has the status and 0 otherwise. We use $|P|$ to denote the number of non-zero entries in the vector P. (Thus, $|P|$ is the number of special items in the dataset.) Our first two constraints (in the formulation given below) require that the distribution of the protected variable be upper and lower bounded. For example, to follow our definition of strong fairness (see Definition 2.1), we would have the constraint $\lfloor \frac{|P|}{k} \rfloor \leq \sum_j p_j z_{i,j} \leq \lceil \frac{|P|}{k} \rceil \ \ \forall i$; in other words, the upper bound $U_i = \lceil \frac{|P|}{k} \rceil$ and the lower bound $L_i = \lfloor \frac{|P|}{k} \rfloor$. In solving ILPs, such inequality constraints are changed to equality constraints using *slack* variables (Schrijver 1998); we use u_i and l_i respectively as the slack variables in the upper and lower bound constraints for $\sum_j p_j z_{i,j}$. In the following specification of constraints, we generalize this to any upper and lower bounds and note they can vary depending on the cluster. The last set of constraints below (i.e., $\sum_i z_{i,j} = 1 \ \ \forall j$) simply require that Z is a valid allocation matrix.

$$\sum_j p_j z_{i,j} + u_i = U_i, \ \ \forall i \tag{1}$$

$$-\sum_j p_j z_{i,j} + l_i = -L_i, \ \ \forall i \tag{2}$$

$$\sum_i z_{i,j} = 1, \ \ \forall j \tag{3}$$

One of the results shown in (Davidson et al. 2022) is the following.

Result #1: *The constraint matrix for the MCMF problem is totally unimodular. Hence, the MCMF problem can be solved in polynomial time for any linear objective function using an efficient linear programming solver.*

For a definition of a totally unimodular matrix and why integer linear programs where the constraint matrix is totally unimodular can be solved in polynomial time, we refer the reader to (Schrijver 1998).

(b) Constraints to enforce group-level fairness: For many data sets, **instance-level constraints** are used to guide clustering algorithms towards desirable solutions (Wagstaff and Cardie 2000). As an example, the **must-link** (ML) constraint $\text{ML}(x, y)$ requires that instances x and y must be placed in the same cluster. In the context of fairness, ML constraints can be used to enforce group-level fairness (Davidson et al. 2022). The feasibility problem with ML constraints (i.e., is there a k-block clustering that satisfies all the given ML constraints?) can be solved efficiently (Davidson and Ravi 2007). However, as the following result indicates, the situation changes completely when one requires a clustering that satisfies strong fairness and the given ML constraints (Davidson et al. 2022).

Result #2: *The problem of determining whether a data set can be partitioned into k clusters that satisfy strong fairness and ML constraints is* ***NP****-complete.*

It should be noted that Result #2 holds even when there is no requirement regarding the quality of the clustering.

3 Detecting Unfairness in Outlier Detection and Clustering

This section focuses on the problem of auditing the output of outlier detection and clustering algorithms to detect unfairness. Here, we assume that there is a set of PSVs. Our work searches for over/under-represented PSV combinations denoted by **x** (which represent groups of individuals). To tie our work back to classic set cover formulations (Garey and Johnson 1979) in theoretical computer science, we formulate our work as searching for a minimum number of occurrences of a disjunction of PSVs (e.g., `Male` $\vee$ `Young`) that is an over-represented in a class compared to the other classes (e.g., in the rest of the population). We search across all PSV combinations (groups of people) to find examples of unfairness. If no such PSV combination is returned, then we conclude that the division of people into classes is fair. A domain expert can determine whether the type of unfairness found is acceptable (or interesting), and our formulations can be run again to explicitly avoid finding such examples of unfairness. For brevity, we discuss one form of unfairness, namely **count-based** unfairness; for other forms of unfairness, we refer the reader to our paper (Davidson and Ravi 2022).

Count-based unfairness: This applies a rule similar to the traditional definition of statistical parity (Kearns et al. 2018); it requires that the count of instances satisfying a PSV combination **x** (normalized by the class size) in a class is nearly the same as the proportion of the PSV count in the rest of the population. This definition of fairness says that a division is unfair if any class violates this rule. We now discuss how constraints can be used to specify such fairness criteria in the context of outlier detection (Davidson and Ravi 2020a). We have extended this approach to clustering in (Davidson and Ravi 2022).

We assume that the PSVs are binary-valued. Given a subset **x** of PSVs, we say that **x covers** an instance y if for at least one of the PSVs in **x**, the value of the corresponding attribute of y is 1. For a given data set D, suppose an outlier detection algorithm outputs D' as the set of outliers. Let α and β be respectively the fraction of instances in D' and $D - D'$ (i.e., the set of normal instances) that are covered by **x**. Let a value $0 < \delta < 1$ be specified by a domain expert as the unfairness threshold. Then, **x** provides a pattern of unfairness if the constraint $|\beta - \alpha| \geq \delta$ is satisfied. An outlier detection algorithm is deemed fair if there is no subset of PSVs with respect to which the output of the outlier detection algorithm is unfair. The following result is shown in (Davidson and Ravi 2022).

Result #3: *The problem of determining whether there is a combination of PSVs that indicates unfairness is* **NP***-hard. The problem can be formulated as an ILP.*

We note that the ILP formulation for this problem needs constraints to capture the notion of coverage and to express the bounds to be satisfied by fraction of instances covered by a subset **x** of PSVs (Davidson and Ravi 2020a). Also, more complex constraints are needed when this formulation is extended to clustering especially when utility values are associated with clusters (Davidson and Ravi 2022).

4 Future Work Involving Fairness and Constraints

We believe that future work on fairness will examine more complex forms of fairness specifications and constraint programming will play an important role in identifying methods to enhance fairness and auditing the outputs of ML algorithms to detect unfairness. We briefly mention some of the possibilities below.

1. That the constraint matrix for disparate impact fairness for a single protected status is totally unimodular begs the questions what other forms of fairness constraints are totally unimodular and how can this be useful?
2. Disparate impact is a very narrow definition of legal fairness but does not completely fulfill the requirement of an ethical algorithm. For example, an algorithm can satisfy disparate impact but can make many inaccurate predictions on protected status individuals. How to efficiently encode different types of fairness as constraints is a critical topic.
3. Our existing work makes a clustering fairer by moving individual points between clusters. More complex forms of modifying a given clustering (e.g., splitting a cluster into multiple new clusters) may be needed to improve fairness. Such modifications change compositions of clusters significantly and some of the modifications may actually reduce the fairness. So, more complex forms of constraints will be needed to specify the allowable set of modifications.
4. In auditing for fairness, more complex notions of coverage may be needed. For example, for a combination **x** of PSVs to cover an instance y, the values of the attributes of y for two or more PSVs may be required to be 1. Also, some applications may need PSVs that can take on more than two integer values, thus permitting more complex coverage requirements. As a consequence, suitable forms of constraints are needed to capture such requirements.
5. There may also be new restrictions/constraints on the combination of PSVs that provide indications of unfairness. For example, one may require such a combination to include or exclude certain subsets of PSVs. Developing suitable specifications of such constraints while ensuring that constraint solvers can scale to large data sets will be very beneficial to practitioners.

Acknowledgment: This work is supported in part by NSF Grants IIS-1908530 and IIS-1910306 titled "Explaining Unsupervised Learning: Combinatorial Optimization Formulations, Methods and Applications".

References

Barocas, S.; Hardt, M.; and Narayanan, A. 2017. Fairness in machine learning. NeurIPS tutorial.

Davidson, I.; Bai, Z.; Tran, C. M.; and Ravi, S. S. 2022. Making Clusterings Fairer by Post-Processing: Algorithms, Complexity Results and Experiments. To appear in *Data Mining and Knowledge Discovery Journal*.

Davidson, I.; and Ravi, S. S. 2007. The complexity of non-hierarchical clustering with instance and cluster level constraints. *Data Min. Knowl. Discov.*, 14(1): 25–61.

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

Davidson, I.; and Ravi, S. S. 2020b. Making Existing Clusterings Fairer: Algorithms, Complexity Results and Insights. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI New York, NY, USA*, 3733–3740. AAAI Press.

Davidson, I.; and Ravi, S. S. 2022. Towards Auditing Unsupervised Learning Algorithms and Human Processes For Fairness. *arXiv preprint arXiv:2209.11762*.

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.

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.

Schrijver, A. 1998. *Theory of Linear and Integer Programming*. John Wiley & Sons.

Wagstaff, K.; and Cardie, C. 2000. Clustering with Instance-Level Constraints. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, 1097–1192.

Xu, R.; and Wunsch, D. C. 2005. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3): 645–678.