Fair Clustering Using Antidote Data

Anshuman Chhabra  
University of California, Davis  
chhabra@ucdavis.edu

Adish Singla  
MPI-SWS  
adishs@mpi-sws.org

Prasant Mohapatra  
University of California, Davis  
pmohapatra@ucdavis.edu

Abstract

Clustering algorithms are widely utilized for many modern data science applications. This motivates the need to make outputs of clustering algorithms fair. Traditionally, new fair algorithmic variants to clustering algorithms are developed for specific notions of fairness. However, depending on the application context, different definitions of fairness might need to be employed. As a result, new algorithms and analysis need to be proposed for each combination of clustering algorithm and fairness definition. Additionally, each new algorithm would need to be reimplemented for deployment in a real-world system. Hence, we propose an alternate approach to group-level fairness in center-based clustering inspired by research on data poisoning attacks. We seek to augment the original dataset with a small number of data points, called antidote data. When clustering is undertaken on this new dataset, the output is fair, for the chosen clustering algorithm and fairness definition. We formulate this as a general bi-level optimization problem which can accommodate any center-based clustering algorithms and fairness notions. We then categorize approaches for solving this bi-level optimization for two different problem settings. Extensive experiments on different clustering algorithms and fairness notions show that our algorithms can achieve desired levels of fairness on many real-world datasets with a very small percentage of antidote data added. We also find that our algorithms achieve lower fairness costs and competitive clustering performance compared to other state-of-the-art fair clustering algorithms.

Keywords: Clustering, Fair Machine Learning, Unsupervised Learning

1. Introduction

With the increasing application of machine learning (ML) algorithms in modern society, the design of fair variants to traditional ML algorithms is an important concern. Vanilla ML algorithms do not account for the biases present in training data against certain minority protected groups, and hence, might reinforce them. Furthermore, clustering has been widely used to find meaningful structures, explanatory underlying processes, generative features, and groupings inherent in a set of examples. It plays a significant role in most modern data science applications, such as in medicine (33), vision (36), language modeling (46), financial decisions (22), and various societal resource allocation problems. Thus, ensuring fairness with respect to protected groups is an important issue for clustering algorithms.

Currently, many different group-level notions for fairness in clustering exist, such as balance (12), proportionality (9), social fairness (20), among others. Traditionally, to make clustering outputs fair with respect to a specific notion of fairness, fair variants to clustering algorithms need to be proposed. Given that many different clustering algorithms exist,
each fair variant proposed requires individual analysis, and possesses different theoretical guarantees. Moreover, if fairness notions or clustering algorithms are changed in a deployed real-world system, the corresponding fair algorithms would also have to be reimplemented.

Therefore, instead of coming up with new fair algorithms for each fairness definition and each clustering algorithm, we propose an alternate approach to ensuring fairness for clustering. Inspired by recent research on adversarial attacks and data poisoning, we aim to augment the dataset with antidote data points such that when we use vanilla clustering on this new combined dataset, fairness constraints are met. Thus, instead of changing the clustering algorithm to ensure fairness, we find an augmented dataset for which the specified fairness constraints are met when vanilla clustering is undertaken on it. Our approach is therefore applicable in very general case scenarios where group-level fairness on the original dataset can be achieved for any arbitrary choice of center-based clustering algorithm and fairness definition. Note that we aim to make clustering fair in the pre-clustering stage as opposed to the in-clustering stage, unlike most research on fair clustering.

Data augmentation to improve fairness was first proposed by (45) for recommendation systems. The authors coined the term antidote data for the data points added to the original dataset. However, since recommendation systems and clustering algorithms differ widely, their problem formulation and techniques do not translate to clustering. The antidote data problem for clustering is then as follows: given a dataset $U$, can we compute (antidote) data $V$ such that when we cluster on $U \cup V$ we obtain a fair clustering output for a chosen fairness notion and clustering algorithm?

We answer this question in the affirmative by proposing a general bi-level formulation of the antidote data problem for clustering. There are also a number of reasons as to why we cannot reuse existing approaches for adversarial attacks on clustering algorithms, which makes our antidote data formulation (and subsequent algorithmic solutions) novel contributions. Firstly, research on adversarial attacks against clustering is sparse, with only two recent papers since 2018 (11; 13). Secondly, these approaches are defined for specific adversarial objectives, and generally aim to change cluster assignments for points near the clustering decision boundary (Theorem 1 in (11)). However, our bi-level formulation requires the antidote data addition to lead to very specific clustering outcomes that improve fairness irrespective of where points lie in clusters. In summary, we make the following contributions:

- We propose an alternative approach to group-level fair clustering, where we augment the original dataset with data points (antidote data) such that when we use vanilla clustering on this new combined dataset, fairness is improved. This is the first work that utilizes data augmentation and antidote points for improving fairness in clustering. In contrast, existing works on fair clustering modify the clustering algorithm specific to a notion of group-level fairness.
- We consider two problem settings for the proposed general bi-level formulation: 1) convex group-level fairness notions and convex center-based clustering objectives, and 2) general group-level fairness notions and general center-based clustering objectives.
- We provide algorithms and analysis for each of these settings, and conduct extensive experiments on real-world datasets for multiple clustering algorithms and fairness notions to demonstrate the efficacy and generality of our approaches.
- We also compare our algorithms to state-of-the-art fair clustering algorithms in terms of fairness, and clustering performance, and find that we achieve improved results on all metrics.

2. Problem Statement

2.1. Proposed Problem

The original dataset is denoted as $U \in \mathbb{R}^{n \times d}$. This is the dataset we wish to augment with some antidote data points such that certain fairness constraints are met when we cluster on the augmented dataset. Furthermore for a matrix $M$, let $M_i$ and $M^i$ denote the $i$-th row and $i$-th column respectively. To start, we first define the clustering problem on $U$. A center-based clustering objective, $C$, takes in a dataset as input (such as $U$) and outputs a set of $k$ centers $\mu \in \mathbb{R}^{k \times d}$, where $k \leq n$. That is, a clustering objective induces a $k$-partition set of the data, where each sample in the dataset is uniquely mapped to a center $\mu_i \in \mu$ where $\mu \in \mathbb{R}^d$. For example, the k-means clustering objective on $U$ can be defined as $C_{\text{k-means}}(U) := \mu = \arg\min_{\mu' \in \mathbb{R}^{k \times d}} \sum_{x \in U} \min_{i \in [k]} ||x - \mu'_i||^2$.

We denote the group-level fairness notion as $F : (\mu, U) \rightarrow \mathbb{R}$. That is, the fairness notion takes as input the set of centers from a clustering algorithm and the original dataset, and outputs a fairness cost. The goal of improving fairness is to then minimize $F$. It is important to note that fairness will be evaluated only on the original real dataset $U$. Moreover, as we will see, all group-level fairness notions can be defined this way.

The General Problem. We now state the antidote data problem for improving fairness. We aim to add a set of data points $V$ to $U$, such that when we cluster on $U \cup V$ and obtain centers $\mu$, $F(\mu, U)$ is less than some given value $\alpha$. The cost of adding points can be defined as the size of set $V$, and hence, we aim to add as few points as possible. The general bi-level optimization problem is as follows:

\[
\begin{align*}
\min_{V, \mu} & \quad |V| \\
\text{s.t.} & \quad F(\mu, U) \leq \alpha \\
& \quad \mu = C(U \cup V)
\end{align*}
\]  

(P1)

Relaxation P1.R. In the paper, we also consider a relaxed formulation of problem P1. This relaxation allows us to propose algorithms that in turn also solve problem P1 indirectly. The idea is to fix the size of the antidote dataset $|V| \leq V_s$ for a given $V_s \in \mathbb{R}$, and optimize the fixed-set $V$ so that we only minimize $F$ in the upper-level problem. Since minimizing the fairness cost is now the upper-level objective, we can also omit writing it as a constraint using $\alpha$:

\[
\begin{align*}
\min_{V, \mu} & \quad F(\mu, U) \\
\text{s.t.} & \quad \mu = C(U \cup V) \\
& \quad |V| \leq V_s
\end{align*}
\]  

(P1.R)

2.2. Definitions

We now define the group-level fairness costs we use in the paper. Consider some $g \in \mathbb{Z}^+$ number of protected groups that comprise $U$. Each protected group has an index $j \in [g]$
and contains a certain number of points of $U$. For simplicity of notation we also assume that a mapping function $\psi(U, j)$ exists which takes in as input $U$ and an integer $j$, where $1 \leq j \leq g$, and gives us the set of points of $U$ which belong to the protected group $j$. Now we can define the social fairness cost of Ghadiri et al (20). This was originally proposed for k-means clustering, but it fits well with any center-based clustering objective where Euclidean distance is used as the clustering distance metric.

**Definition 1 (Social Fairness (20)).** Let $\Delta(\mu, U) = \sum_{x \in U} \min_{\mu_i \in \mu} ||x - \mu_i||^2$ where $U$ is the original dataset and $\mu$ are cluster centers. Then the social fairness cost is defined as:

$$F_{social}(\mu, U) = \max_{j \in [g]} \left\{ \frac{\Delta(\mu, \psi(U, j))}{|\psi(U, j)|} \right\}$$

Next we define the balance metric (12; 3). Traditionally, balance is a fairness metric that is not a cost, and is maximized. To fit within our framework, we frame it as a cost by multiplying it with $-1$, and name it the balance cost. Again, for simplicity of notation, we assume a mapping function $\phi(U, \mu, i)$ exists which takes in as input $U$, $\mu$, and a cluster label $i \in [k]$ and gives us the points in $U$ which belong to cluster $i$. Note that obtaining cluster labels is trivial as for each $x \in U$ the corresponding label can be obtained as $i = \arg\min_{i' \in [k]} ||x - \mu_{i'}||$. 

**Definition 2 (Balance Cost (3)).** Let $U$ be the original dataset and $\mu \in \mathbb{R}^{k \times d}$ be the set of cluster centers. Define the following ratio $R(i, j) = \frac{|\psi(U, j)|/|U|}{|\psi(U, j)|/|\phi(U, \mu, i)|}$ which signifies the ratio between the proportion of points of group $j$ in $U$ and proportion of group $j$ points in cluster $i$. The balance cost $F_{balance} \in [-1, 0]$ is then defined:

$$F_{balance}(\mu, U) = -\min_{i \in [k], j \in [g]} \left\{ \min\left\{ R(i, j), \frac{1}{R(i, j)} \right\} \right\}$$

3. Proposed Approaches

We consider problem P1 under 2 different settings and provide algorithms and analysis for each: (1) Convex $C$ and Convex $F$, and (2) General $C$ and General $F$. While setting (1) comprises more of a toy problem as clustering objectives used in practice are rarely convex, solving problem P1 for setting (2) is quite challenging. For the first setting with convex functions, we can reduce the bi-level problem to a single-level optimization, allowing us to utilize off-the-shelf solvers to obtain $V$. For the general setting, the antidote data problem is significantly harder and we resort to using zeroth-order optimizers as part of our proposed solution to finding a feasible $V$.

3.1. Convex $C$ and Convex $F$

For this setting, we assume that both $C$ and $F$ are convex functions. Assuming convexity allows us to effectively reduce the bi-level problem to a single-level form, which can then be provided to off-the-shelf convex/non-convex solvers for optimization. In particular, we exploit the convexity of the functions by replacing the lower-level problem with its Karush-Kuhn-Tucker (KKT) optimality conditions as constraints for the upper-level problem. Since
the lower-level clustering problem is convex, the KKT conditions are necessary and sufficient to ensure optimality (16).

As optimizing bi-level problems is in general NP-Hard (50), and problem P1 contains an NP-Hard cardinality minimization problem (1) as the upper-level objective, we use the relaxed form P1.R to indirectly solve P1. This involves fixing |V| as an input hyperparameter and optimizing V so as to minimize $F$, without considering $\alpha$. We then use the convexity of the lower-level problem to obtain a single-level reduction from this bi-level problem by replacing the lower-level problem with its KKT constraints. When we minimize this reduced single-level problem, we effectively minimize P1.R.

We describe our approach as Algorithm 1. We aim to solve problem P1.R using our algorithm, and in each iteration try to find a suitable V to optimize using the reduced single-level problem (obtained via KKT conditions). In each iteration of the algorithm, we start by fixing the size of V to some $V^s$, and obtain $F$ after optimizing V. If this fairness cost is less than $\alpha$, we can exit, otherwise we increase the size of V (denoted as $V^s$) by $\xi \in \mathbb{Z}^+$ for the next iteration and continue. Algorithm 1 can also exit if the constraint is not met, if a certain number of iterations are exceeded, or if |V| grows to an unacceptable value. We omit these details from Algorithm 1 for simplicity, but they can be easily implemented.

Not many widely used convex formulations for clustering algorithms exist except for sum-of-norms (SON) clustering (34; 26), which is strongly convex. SON clustering has been shown to be a convex relaxation to both k-means clustering (34) and hierarchical agglomerative clustering (26). Below, we analyze SON clustering in the context of Algorithm 1. For the fairness notion, we utilize $F_{social}$ which is clearly convex and well-defined for SON clustering. We first define the SON clustering objective. It is important to note that we modify the notation– since the objective is convex, the number of clusters are not discretely defined, but obtained via a regularization parameter $\lambda$. Centers are represented as a $\mathbb{R}^{n \times d}$ matrix as there is no explicitly defined k, but note there will only be some unique $k \leq n$ centers decided by the parameterization of $\lambda$. The objective is as follows:

$$C_{SON}(U) := \mu = \underset{\mu \in \mathbb{R}^{n \times d}}{\text{argmin}} \frac{1}{2} \sum_{j=1}^{n} ||U_j - \mu_j||^2 + \lambda \sum_{i<j} ||\mu_i - \mu_j||.$$ 

Let $V_s(t)$ denote the size $V_s$ of V in iteration t of Algorithm 1 (line 2). The number of centers we have will be $\mu \in \mathbb{R}^{m \times d}$ where $m = n + V_s(t)$ for $U \cup V$. To derive the KKT conditions we first reformulate the objective. Consider an ordering of all $(\mu_i, \mu_j)$ pairs where all $i < j$. We can let each of the m centers $\mu_i$ be a node in a graph $G$. The created ordering essentially enumerates the list of edges E for the graph $G$. We denote this ordering as $O$ where we will have |E| = |O| = m(m − 1)/2. We also denote the node-arc-incidence matrix (30) for $(G, E)$ as $I \in \mathbb{R}^{m \times |O|}$. We can then rewrite the SON objective, define the dual problem to the reformulation, and derive the KKT conditions (details provided in Section A.2 of appendix). Then the single-level reduction for P1.R can be written as follows:

$$\min_{V, \mu, \eta, \theta, \zeta} F_{social}(\mu, U)$$

s.t. $\theta + \mu - (U \cup V) = 0$

$\eta - \max\{0, 1 - (1/||\eta + \zeta||)\} (\eta + \zeta) = 0$

$\mu^T I - \eta = 0$

$I \zeta^T - \theta = 0$
Here, $\mu \in \mathbb{R}^{m \times d}, \eta \in \mathbb{R}^{d \times |O|}$ are the primal variables, and $\theta \in \mathbb{R}^{m \times d}, \zeta \in \mathbb{R}^{d \times |O|}$ are the dual variables. We also observe that replacing KKT conditions as constraints can introduce non-convexity. All the constraints and objectives are convex, except for one:

$$\eta - \max\{0, 1 - (1/||\eta + \zeta||)(\eta + \zeta)\} = 0.$$ 

To approximate this, we can replace it with an affine constraint as $\eta - \gamma (\eta + \zeta) = 0$ where $0 \leq \gamma \leq 1$. Then a convex solver such as CVX (18) can be used to solve the above problem. Finally, assuming it takes time $T_{\text{KKT}}$ to solve the single-level problem, and a feasible antidote dataset $V^*$ exists, Algorithm 1 has a running time of $O(T_{\text{KKT}}|V^*|/\xi)$.

**Remark.** Since we are solving a convex problem above, the results for this setting are not too difficult to obtain. We thus defer results for Algorithm 1 to the appendix (Section B).

Algorithm 1: Convex $C$ and $F$

Input: $U, C, F, V_s, \xi$

Output: $V$

1. while true do
2. initialize $V$ arbitrarily with $|V| = V_s$
3. reduce problem $P1.R$ by replacing $C(U \cup V)$ with its KKT conditions as constraints
4. solve this single-level problem for optimal $V$
5. if $F(\mu, U) \leq \alpha$ return $V$ else $V_s \leftarrow V_s + \xi$
6. end

Algorithm 2: General $C$ and $F$

Input: $U, C, F, A, V_s, n', \xi$

Output: $V$

1. while true do
2. define $\mu \leftarrow C(U \cup V)$ and $f(V) \leftarrow F(\mu, U)$
3. initialize $V$ arbitrarily with $|V| = V_s$
4. optimize $V$ using SRE($n', f(V), A$)
5. obtain optimized $V$ and $F(\mu, U)$ from SRE & $A$
6. if $F(\mu, U) \leq \alpha$ return $V$ else $V_s \leftarrow V_s + \xi$
7. end

3.2. General $C$ and General $F$

In this setting, we make no assumptions about the clustering objective $C$ and the fairness cost $F$. In such a minimal assumption setting where group-level fairness notions as well as center-based clustering objectives can vary widely, it is not trivial to propose algorithms with strong theoretical guarantees. Furthermore, some of the most popular and widely utilized clustering algorithms such as k-means, hierarchical clustering, DBSCAN, etc. possess highly non-convex objectives and are generally optimized via heuristic algorithms (such as Lloyd’s algorithm for k-means). In terms of fairness notions for clustering, balance is generally the
most widely used metric in proposing fair algorithms. As evident in Definition 2, it is both non-convex and non-differentiable.

Furthermore, general bi-level optimization is NP-Hard; even for the simpler case when the upper-level and lower-level problems are linear, a polynomial time algorithm that finds the global optima of the bi-level problem might not exist (50). Since we are dealing with possibly many non-convex upper-level and lower-level problems in this setting, finding a global optima for P1 is not a trivial task. We then resort to finding a locally optimal solution that satisfies our problem constraints. To do this, we relax the NP-Hard upper-level problem which seeks to minimize the size of the antidote dataset \( V \). Similar to the convex setting, we are attempting to solve the relaxed formulation \( P1.R \) (indirectly solving \( P1 \)), where we fix \(|V|\) to some given value, and optimize \( V \) to minimize \( \mathcal{F} \).

To solve \( P1.R \), we can use zeroth-order optimization algorithms (such as RACOS (55), CMAES (23), IMGPO (29)). Let such an algorithm be denoted as \( \mathcal{A} \). Most zeroth-order optimization algorithms do not scale well with problem input, and hence, cannot usually be applied to data with number of samples \( n \geq 1000 \) (42). However, since our goal is to utilize antidote data on large-scale datasets, the algorithm \( \mathcal{A} \) cannot be applied directly to solve \( P1.R \) in practice. To circumvent this problem, we propose using the Sequential Random Embedding (SRE) approach of (42), which can be used in conjunction with the zeroth-order blackbox optimizer \( \mathcal{A} \) to solve \( P1.R \). The SRE approach scales the problem input by projecting it to a low-dimensional setting where it invokes \( \mathcal{A} \) to solve the optimization. SRE takes in as input the reduced dimension \( n' \ll n \), the objective function \( f \) to optimize, and zeroth-order optimization algorithm \( \mathcal{A} \). We defer the reader to (42) for more details on SRE.

Using the SRE approach, we propose Algorithm 2 for solving \( P1.R \). We begin by defining the nested function \( f \) to optimize (line 2) which takes in as input some \( V \) and outputs the fairness cost \( \mathcal{F}(\mu, U) \) where \( \mu \) is obtained via \( \mathcal{C}(U \cup V) \). The basic idea is to fix \(|V|\) to some pre-defined starting value \( V_s \) and optimize \( V \) using the SRE approach as the back-end (line 3-5). Then, if the constraint \( \mathcal{F}(\mu, U) \leq \alpha \) is not met, we increase \(|V|\) by some small number \( \xi \in \mathbb{Z}^+ \) and repeat (line 6). Similar to Algorithm 1, we can exit in the while loop after a certain number of iterations or if \(|V| \gg |U|\).

In our experiments for this setting, we use RACOS (55) as the algorithm \( \mathcal{A} \), which is a Sampling-and-Learning (SAL) framework. Previous work on SAL approaches allows us to give some weak theoretical results regarding Algorithm 2 on computing a locally optimal solution for Problem \( P1.R \) and the number of blackbox queries required to do so. We present Theorem 3, which we have adapted from (54) for our setting. Essentially the result states that the query complexity to compute a locally optimal solution given a fixed-size \( V \) to optimize, scales inversely with how effectively \( \mathcal{A} \) samples feasible solutions and how many feasible solutions \( f \) admits. This does not provide much information from a practical perspective, however through experiments we obtain competitive results on real-world datasets for different combinations of \( \mathcal{F} \) and \( \mathcal{C} \). Finally, if \( \mathcal{A} \) runs for time \( T_A \), and assuming a feasible antidote dataset \( V^* \) exists, Algorithm 2 has a running time of \( O(T_A|V^*|/\xi) \).

**Theorem 3** (54). Let \( V^* \in \mathbb{R}^{V_s(t)}_{+} \times d \) be a minimizer for the function \( f(V) \) in an iteration \( t \) of Algorithm 2 and for \( \epsilon > 0 \) define \( X = \{ V \in \mathbb{R}^{V_s(t)}_{+} \times d \mid f(V) - f(V^*) \leq \epsilon \} \). Let \( \mathbb{P}_X \) denote the average probability of successfully sampling from the uniform distribution over \( X \) by algorithm \( \mathcal{A} \), and it takes \( n_X \) samples to realize \( \mathbb{P}_X \). Then, the number of queries to \( f \) that
A makes to compute $\tilde{V}$ s.t. $f(\tilde{V}) - f(V^*) \leq \epsilon$ with probability at least $1 - \delta$ is bounded as $O(\max\{\frac{\ln(\delta^{-1})}{\epsilon}, n_X\})$.

4. Results

4.1. Datasets

We consider four real-world datasets commonly used to evaluate fair clustering algorithms: adult (32), bank (40), creditcard (53), and Labeled Faces in the Wild (LFW) (27). The adult dataset has 10000 × 5 samples, and protected groups signify race (white, black, asian-pac-islander, amer-indian-eskimo, other). The bank dataset has 45211 × 3 samples, and protected groups signify marital status (married, single, divorced). The creditcard dataset has 30000 × 23 samples, and the protected groups signify education (higher and lower education). LFW has 13232 × 80 samples, and the protected groups signify sex (male, female).

We defer the results for Algorithm 1 (with $C_{\text{SON}}$ and $F_{\text{social}}$) to the appendix (Section B) as we are solving a convex problem for which the results can be obtained in a straightforward manner.

4.2. Results for Algorithm 2

We compare Algorithm 2 against vanilla clustering and state-of-the-art fair clustering algorithms. Throughout we let $k = 2$ and due to space limitations, present results for $k = 3$ and $k = 4$ in the appendix (Section C.1). We also compare Algorithm 2 and other fair clustering approaches in terms of clustering performance, using clustering performance metrics such as the Silhouette coefficient (47), Calinski-Harabasz score (7), and the Davies-Bouldin index (15). We use these metrics to unify comparisons across the different clustering algorithms considered in experiments. For all experiments, we choose $\alpha$ to be the fairness cost of the algorithms being compared against (vanilla clustering, fair algorithms) so as to improve on them. We let $\mathcal{A}$ be the RACOS (55) algorithm, $V_s = 10, n' = 100, \xi = 1$.

4.2.1. Comparing Algorithm 2 With Vanilla Clustering and Fair Clustering Approaches

Since Algorithm 2 can accommodate general $C$ and $F$, we experiment on 3 combinations: Combination #1 with $C_{\text{k-means}}$ and $F_{\text{balance}}$, Combination #2 with $C_{\text{k-means}}$ and $F_{\text{social}}$, and Combination #3 where $C$ is unnormalized spectral clustering, and $F$ is $F_{\text{balance}}$. The results when comparing against vanilla clustering are shown in Table 1. Vanilla cluster centers are denoted as $\mu^{\text{vanilla}}$ and centers obtained via Algorithm 2 are denoted by $\mu$. As can be seen we add very few antidote data points ($|V|/|U|$) and improve on the fairness cost over vanilla clustering. For each of the combination settings considered, we also compare against an equivalent state-of-the-art fair clustering algorithm. For Combination #1 we consider the algorithm of Bera et al (3), for Combination #2 we consider the Fair-Lloyd algorithm of Ghadiri et al (20), and for Combination #3 we consider the algorithm of Kleindessner et al (31). Since the approach of (31) cannot handle large datasets, we subsample each dataset to 1000 samples for Combination #3. The results are shown in Table 2, and centers obtained from fair clustering algorithms are denoted as $\mu^{\text{SOTA}}$. We find that we outperform fair algorithms in terms of lower fairness costs.
Table 1: Comparing fairness costs of Algorithm 2 with vanilla clustering. (Consider Combination #1 and the bank dataset as an example. The fairness cost for the vanilla cluster centers $\mu^{\text{vanilla}}$ is $F(\mu^{\text{vanilla}}, U) = -0.3054$ and $\alpha$ is set to this value to improve on this fairness cost. After Algorithm 2 is run, $V$ is obtained, with size $|V| = 0.00011|U|$. Cluster centers $\mu$ obtained by clustering on $U \cup V$ result in fairness cost $F(\mu, U) = -0.3077$. This is lower than $F(\mu^{\text{vanilla}}, U)$, leading to improved fairness. Refer to Section 4.2 for more details.)

| Clustering-Fairness Combination | Dataset | $\alpha$ | $|V|/|U|$ | $F(\mu^{\text{vanilla}}, U)$ | $F(\mu, U)$ |
|-------------------------------|---------|---------|-------|-----------------|--------------|
| Combination #1: $C_k$-means, $F_{\text{balance}}$ | adult    | -0.6199 | 0.001 | -0.619         | -0.6196     |
|                               | bank     | -0.3054 | 0.00011 | -0.3054       | -0.3077     |
|                               | creditcard | -0.8696 | 0.00017 | -0.8696       | -0.8715     |
|                               | LFW      | -0.8815 | 0.00075 | -0.8815       | -0.8821     |
| Combination #2: $C_k$-means, $F_{\text{social}}$ | adult    | 5.3678  | 0.0005 | 5.3678         | 4.2104      |
|                               | bank     | 2.3432  | 0.00022 | 2.3432        | 2.3416      |
|                               | creditcard | 19.740  | 0.00034 | 19.740        | 19.729      |
|                               | LFW      | 1406.3411 | 0.00076 | 1406.3411     | 1406.1676   |
| Combination #3: $C_{\text{spectral}}, F_{\text{balance}}$ | adult    | -0.4811 | 0.00222 | -0.4811       | -0.5489     |
|                               | bank     | -0.8384 | 0.00034 | -0.8384       | -0.8407     |
|                               | creditcard | -0.9279 | 0.00076 | -0.9279       | -0.9389     |
|                               | LFW      | -0.9279 | 0.00076 | -0.9279       | -0.9389     |

Table 2: Comparing fairness costs of Algorithm 2 with fair clustering algorithms. (Reads similarly to Table 1.)

| Clustering-Fairness Combination | Dataset | $\alpha$ | $|V|/|U|$ | $F(\mu^{\text{SOTA}}, U)$ | $F(\mu, U)$ |
|-------------------------------|---------|---------|-------|-----------------|--------------|
| Combination #1: $C_k$-means, $F_{\text{balance}}$ | adult    | -0.6059 | 0.001 | -0.6059        | -0.6196     |
|                               | bank     | -0.3065 | 0.00011 | -0.3065       | -0.3077     |
|                               | creditcard | -0.8696 | 0.00017 | -0.8696       | -0.8715     |
|                               | LFW      | -0.8816 | 0.00075 | -0.8816       | -0.8821     |
| Combination #2: $C_k$-means, $F_{\text{social}}$ | adult    | 4.2636  | 0.0005 | 4.2636         | 4.2104      |
|                               | bank     | 2.3135  | 0.1549 | 2.3135         | 2.3119      |
|                               | creditcard | 18.998  | 0.19   | 18.998         | 18.998      |
|                               | LFW      | 1344.5468 | 0.3999 | 1344.5468     | 1344.5461   |
| Combination #3: $C_{\text{spectral}}, F_{\text{balance}}$ | adult    | -0.5973 | 0.001 | -0.5973       | -0.6911     |
|                               | bank     | -0.6086 | 0.5    | -0.6086       | -0.6899     |
|                               | creditcard | -0.8407 | 0.38   | -0.8407       | -0.9990     |
|                               | LFW      | -0.9926 | 0.4    | -0.9926       | -0.9997     |

4.2.2. COMPARING CLUSTERING PERFORMANCE

For comparison, we use the widely utilized Silhouette score (47) which lies between $[-1, 1]$, with higher scores indicating better clustering performance. We show the results in Figure 1 for each combination setting considered. The fair clusters of Algorithm 2 used here are the same from Table 1. We observe that despite outperforming fair algorithms in terms of fairness, we still exhibit competitive clustering performance. We defer the results for the other performance metrics to the appendix (Section C.2), since those are unbounded and harder to interpret.

5. Related Works

**Fairness in Machine Learning.** ML algorithms can be made fair in three stages of the learning pipeline (8; 39)– before-training (pre-processing the dataset), during-training (changing the ML algorithm), or after-training (post-processing the learnt model). Most
research on fair clustering focuses on the \textit{during-training} phase (3; 4; 2; 12; 48; 61; 31; 20) and proposes fair clustering algorithms. In their paper, (14) study the \textit{after-training} phase for improving fairness post-clustering. The approaches proposed in our paper are novel since they improve fairness for clustering models in the \textit{before-training} stage. Further, our approaches can accommodate general fairness notions and clustering algorithms.

\textbf{Machine Teaching.} Our approach in this paper is inspired by the techniques in \textit{machine teaching} literature (21; 19; 49; 59; 60; 10; 41). Machine teaching studies the interaction between a teacher and a learner where the teacher selects training examples for the learner to learn a specific task. A machine teaching problem can be cast in a bi-level form where the upper-level problem defines the teacher’s cost and the lower-level problem defines the learner’s method. Variations of this bi-level form can be used to formulate teacher’s optimization problem in a variety of learning settings, including supervised learning (58; 35; 38; 17), imitation learning (6; 25; 28; 5; 51), and reinforcement learning (56; 57; 43; 37; 44). In the proposed antidote data problem for clustering, the upper-level problem (teacher’s cost) is the cost of adding antidote data, and the lower-level problem (learner) is the clustering algorithm.

\textbf{Bi-level Optimization.} Bi-level problems involve a two-level hierarchical optimization. For these, a lower-level problem exists, which influences the solutions for an upper-level problem. Both bi-level optimization and verifying the optimality of an obtained solution are NP-Hard (24; 52). This makes finding optimal solutions and evaluating them non-trivial tasks. In the paper, the main problem considered is a complex bi-level optimization, where both upper-level and lower-level problems can be non-convex optimization problems. Many techniques for bi-level programming exist, but most of these assume simple forms for the upper/lower problems, or use evolutionary methods for which theoretical results are hard to provide (50). Despite these challenges, we provide algorithms that obtain feasible solutions to the bi-level problem and outperform state-of-the-art fair clustering approaches.

6. Concluding Discussions

We propose the antidote data problem for improving group-level fairness in center-based clustering. We provide a more general alternative to traditional approaches aimed at making clustering fair. Instead of proposing new fair variants to clustering algorithms, we augment
the original dataset with new *antidote* data points. When regular clustering is undertaken on this new dataset, the clustering output is *fair*. This approach inspired by research on data poisoning attacks, voids the need to come up with new fair algorithms or individual analysis, for different group-level fairness notions or center-based clustering algorithms. Our approach also does not require reimplementation for deployment in actual systems, if the fairness notion or clustering algorithm is changed. We find that our algorithms only need to add a small percentage of points to achieve the given fairness constraints on many real-world datasets without loss of clustering performance.

A major limitation of our work is running time. While not prohibitively slow, in comparison to fair clustering algorithms, Algorithm 2 is generally slower and requires careful parameterization for convergence. Similar limitations hold for the other algorithm. However, we believe that despite these shortcomings, our paper opens up an important alternative direction for future research in fair clustering, as our experiments also demonstrate. For future work, we aim to provide faster and more general algorithms for the bi-level problem.

References


