Fair Algorithms for Clustering

Suman K. Bera* Deeparnab Chakrabarty† Nicolas J. Flores‡ Maryam Negahbani§

Abstract

We study the problem of finding low-cost *fair clusterings* in data where each data point may belong to many protected groups. Our work significantly generalizes the seminal work of Chierichetti et al. (NIPS 2017) as follows.

- We allow the user to specify the parameters that define fair representation. More precisely, these parameters define the maximum over- and minimum under-representation of any group in any cluster.
- Our clustering algorithm works on any ℓ_p-norm objective (e.g. k-means, k-median, and k-center). Indeed, our algorithm transforms any vanilla clustering solution into a fair one incurring only a slight loss in quality.
- Our algorithm also allows individuals to lie in multiple protected groups. In other words, we do not need the protected groups to partition the data and we can maintain fairness across different groups simultaneously.

Our experiments show that on established data sets, our algorithm performs much better in practice than what our theoretical results suggest.

1 Introduction

Many important decisions today are made by machine learning algorithms. These range from showing advertisements to customers [59, 31], to awarding home loans [47, 56], to predicting recidivism [10, 32, 27]. It is important to ensure that such algorithms are *fair* and are not biased towards or against some specific groups in the population. A considerable amount of work [46, 68, 25, 45, 20, 67, 66] addressing this issue has emerged in the recent years.

Our paper considers fair algorithms for clustering. Clustering is a fundamental unsupervised learning problem where one wants to partition a given data-set. In machine learning, clustering is often used for feature generation and enhancement as well. It is thus important to consider the bias and unfairness issues when inspecting the quality of clusters. The question of fairness in clustering was first asked in the beautiful paper of Chierichetti et al. [25] with subsequent generalizations by Rösner and Schmidt [60].

In this paper, we give a much more generalized and tunable notion of fairness in clustering than that in [25, 60]. Our main result is that any solution for a wide suite of vanilla clustering objectives can be transformed into fair solutions in our notion with only a slight loss in quality by a simple algorithm.

Many works in fairness [20, 25, 60, 19] work within the disparate impact (DI) doctrine [36]. Broadly speaking, the doctrine posits that any "protected class" must have approximately equal representation in the

*UC Santa Cruz, Email: sbera@ucsc.edu
†Dartmouth College, Email: deeparnab@dartmouth.edu
‡Dartmouth College, Email: nicolasflores.19@dartmouth.edu
§Dartmouth College, Email: maryam@cs.dartmouth.edu

decisions taken (by an algorithm). Although the DI doctrine is a law [1, 35] in the United States, violating the DI doctrine is by itself *not* illegal [5]; it is illegal only if the violation cannot be justified by the decision maker. In the clustering setting, this translates to the following algorithmic question : what is the loss in quality of the clustering when all protected classes are required to have approximately equal representation in the clusters returned?

Motivated thus, Chierichetti et al. [25], and later Rösner and Schmidt [60], model the set of points as partitioned into ℓ colors, and the color proportion of each returned cluster should be similar to that in the original data. There are three shortcomings of these papers: (a) the fairness constraint was too stringent and brittle, (b) good algorithms were given only for the k-center objective, and (c) the color classes weren't allowed to overlap. We remark that the last restriction is limiting since an individual can lie in multiple protected classes (consider an African-American senior woman). In our work we address all these concerns: we allow the user to specify the fairness constraints, we give simple algorithms with provable theoretical guarantees for a large suite of objective functions, and we allow overlapping protected classes.

Our fairness notion. We propose a model which extends the model of [25] to have $\ell \geq 2$ groups of people which are allowed to overlap. For each group i, we have two parameters $\beta_i, \alpha_i \in [0, 1]$. Motivated by the DI doctrine, we deem a clustering solution *fair* if each cluster satisfies two properties: (a) *restricted dominance (RD)*, which asserts that the fraction of people from group i in any cluster is at most α_i, and (b) *minority protection (MP)*, which asserts that the fraction of people from group i in any cluster is at least β_i. Note that we allow β_i, α_i's to be arbitrary parameters, and furthermore, they can differ across different groups. This allows our model to provide a lot of flexibility to users. For instance, our model easily captures the notions defined by [25] and [60].

We allow our protected groups to overlap. Nevertheless, the quality of our solutions depend on the amount of overlap. We define Δ (similar to [20]) to be the maximum number of groups a single individual can be a part of. This parameter, as we argued above, is usually not 1, but can be assumed to be a small constant depending on the application.

Our results. Despite the generality of our model, we show that *in a black-box fashion*, we can get fair algorithms for *any* ℓ_p-norm objective (this includes, k-center, k-median, and the widely used k-means objective) if we allow for very small additive violations to the fairness constraint. We show that *given any ρ-approximation algorithm $\mathcal{A}$ for a given objective which could be returning widely unfair clusters, we can return a solution which is a $(\rho + 2)$-approximation to the best clustering which satisfies the fairness constraints* (Theorem 1). Our solution, however, can violate both the RD and MP property *additively* by $4\Delta + 3$. This is negligible if the clusters are large, and our empirical results show this almost never exceeds 3. Further in our experiments, our cost is at most 15% more than optimum, which is a much better factor compared to $(\rho + 2)$.

The black-box feature of our result is useful also in *comparing* the performance of any particular algorithm $\mathcal{A}$. This helps if one wishes to justify the property of an algorithm one might be already using. Our results can be interpreted to give a way to convert any clustering algorithm to its fair version. Indeed, our method is very simple – we use the solution returned by $\mathcal{A}$ to define a *fair assignment* problem and show that this problem has a good optimal solution. The fair assignment problem is then solved via iterative rounding which leads to the small additive violations. In the case of $\Delta = 1$ (disjoint groups), we can get a simpler, one-iteration rounding algorithm.

Finally, we show that our simple approach also leads to algorithms for a related clustering problem. In many clustering applications involving anonymity and privacy [4, 60], one requires the size of the cluster to be *at least* a certain size L. We show that *given any ρ-approximation for the vanilla clustering problem in any ℓ_p norm, we can get a $(\rho+2)$-approximation for the lower bounded clustering problem in* $O(2^k \mathrm{poly}(n))$

time (Theorem 2). Thus, our algorithm is a *fixed-parameter tractable (FPT)* approximation algorithm and in particular, armed with the recent result of Cohen-Addad et al. [28], implies a 3.736-factor approximation algorithm for the lower bounded k-median problem in $k^{O(k)}\mathrm{poly}(n)$ time. To put this in perspective, in polynomial time one can only get a large $O(1)$-approximation (see footnote 5). Furthermore, for higher norms, no constant factor approximations are known.[1]

Comparison with recent works. In a very recent independent and concurrent work, Schmidt et al. [61] consider the fair k-means problem in the streaming model with a notion of fairness similar to ours. However, their results crucially assume that the underlying metric space is Euclidean. Their main contributions are defining "fair coresets" and showing how to compute them in a streaming setting, resulting in significant reduction in the input size. Although their coreset construction algorithm works with arbitrary number of groups, their fair k-means algorithms assume there are only two disjoint groups of equal size. Even for this, Schmidt et al. [61] give an $(5.5\rho + 1)$-approximation, given any ρ-approximation for the vanilla k-means problem; the reader should compare with our $(\rho+2)$-approximation. Backurs et al. [12] consider the problem of designing scalable algorithm for the fair k-median problem in the Euclidean space. The notion of fairness is *balance*, as defined by Chierichetti et al. [25], and hence works only for two disjoint groups. Their approximation ratio is $O_{r,b}(d \log n)$ where r and b are fairness parameters, and d is the dimension of the Euclidean space. In contrast, our fair k-means and k-median algorithms works in any metric space, with arbitrary number of overlapping groups.

In another independent and parallel work, Bercea et al. [14] consider a fairness model that is similar to ours. They give a similar, but arguably more complicated algorithm for a variety of clustering objectives. Ahmadian et al. [6] study the k-center objective with only *restricted dominance (RD)* type constraints and give bi-criteria approximations. In comparison, we emphasize on a simple, yet powerful unifying framework that can handle any ℓ_p-norm objective. None of the above works handle overlapping groups.

1.1 Other related works

Fairness in algorithm design has received a lot of attention lately [17, 55, 34, 36, 46, 68, 25, 45, 20, 67, 66, 20, 19, 29, 50, 37]. Our work falls in the category of designing fair algorithms, and as mentioned, we concentrate on the notion of *disparate impact*. Feldman et al. [36] and Zafar et al. [67] study the fair classification problem under this notion. Celis et al. in [20], Celis et al. in [19], and Chierichetti et al. in [26] study respectively the fair ranking problem, the multiwinner voting problem, and the matroid optimization problem; All of these works model fairness through *disparate impact*. Chierichetti et al. in [25] first addresses *disparate impact* for clustering problems in the presence of two groups, Rösner and Schmidt [60] generalizes it to more than two groups.

Chen et al. [24] define a notion of proportionally fair clustering where all possible groups of reasonably large size are entitled to choose a center for themselves. This work builds on the assumption that sometimes the task of identifying protected group itself is untenable. Kleindessner et al. in [51] study the problem of enforcing fair representation in the data points chosen as cluster center. This problem can also be posed as a matroid center problem. Kleindessner et al. in [52] extends the fairness notion to graph spectral clustering problems. Celis et al. in [18] proposes a meta algorithm for the classification problem under a large class of fairness constraints with respect to multiple non-disjoint protected groups.

Clustering is a ubiquitous problem and has been extensively studied in diverse communities (see [3] for a recent survey). We focus on the work done in the algorithms and optimization community for clustering problems under ℓ_p norms. The $p = \{1, 2, \infty\}$ norms, that is the k-median, k-means, and k-center problems

[1] For the special case of Euclidean k-means, there are PTASes in [30, 15] with run times exponential in k.

respectively, have been extensively studied. The k-center problem has a 2-approximation [40, 38] and it is NP-hard to do better [41]. A suite of algorithms [23, 44, 22, 13, 54] for the k-median problem has culminated in a 2.676-approximation [16], and is still an active area of research. For k-means, the best algorithm is a $9 + \varepsilon$-approximation due to Ahmadian et al. [7]. For the general p-norm, most of the k-median algorithms imply a constant approximation.

Capacitated clustering is similar to fair clustering in that in both, the assignment is not implied by the set of centers opened. We already mentioned the results for *lower bounded* clustering. One can also look at *upper bounded* clustering where every cluster is at most a size U. The (upper-bounded) *capacitated k-median* problem is one of the few classic problems remaining for which we do not know $O(1)$-approximations, and neither we know of a good hardness. The capacitated k-center problem has a 6-approximation [48]. Recently, an FPT algorithm was designed by [2]; They show a $7 + \varepsilon$-approximation for the upper bounded capacitated k-median problem which runs in time $O(f(k) \cdot \mathrm{poly}(n))$ where $f(k) \sim k^{O(k)}$. It is instructive to compare this with our result on lower bounded k-median problem.

2 Preliminaries

Let C be a set of points (whom we also call "clients") we want to cluster. Let these points be embedded in a metric space $(\mathcal{X}, d)$. We let $F \subseteq \mathcal{X}$ be the set of possible cluster center locations (whom we also call "facilities"). Note F and C needn't be disjoint, and indeed F could be equal to C. For a set $S \subseteq \mathcal{X}$ and a point $x \in \mathcal{X}$, we use $d(x, S)$ to denote $\min_{y \in S} d(x, y)$. For an integer n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

Given the metric space $(\mathcal{X}, d)$ and an integer parameter k, in the VANILLA (k,p)-CLUSTERING problem the objective is to (a) *"open"* a subset $S \subseteq F$ of at most k facilities, and (b) find an *assignment* $\phi : C \to S$ of clients to open facilities so as to minimize $\mathcal{L}_p(S; \phi) := \left(\sum_{v \in C} d(v, \phi(v))^p\right)^{\frac{1}{p}}$. Indeed, in this vanilla version with no fairness considerations, every point $v \in C$ would be assigned to the closest center in S. The case of $p = \{1, 2, \infty\}$, the k-median, k-means, and k-center problems respectively, have been extensively studied in the literature [40, 38, 23, 44, 22, 13, 54, 16, 7]. Given an instance $\mathcal{I}$ of the VANILLA (k,p)-CLUSTERING problem, we use $\mathrm{OPT}_{\mathsf{vnll}}(\mathcal{I})$ to denote its optimal value.
The next definition formalizes the fair clustering problem which is the main focus of this paper.

Definition 1 (FAIR (k,p)-CLUSTERING Problem)**.** In the fair version of the clustering problem, one is additionally given ℓ many (not necessarily disjoint) *groups* of C, namely $C_1, C_2, \ldots, C_\ell$. We use Δ to denote the maximum number of groups a single client $v \in C$ can belong to; so if the C_j's were disjoint we would have $\Delta = 1$. One is also given two *fairness vectors* $\vec{\alpha}, \vec{\beta} \in [0,1]^\ell$.

The objective is to (a) *open* a subset of facilities $S \subseteq F$ of at most k facilities, and (b) find an *assignment* $\phi : C \to S$ of clients to the open facilities so as to minimize $\mathcal{L}_p(S; \phi)$, where ϕ satisfies the following *fairness constraints*.

$$\Big|\{v \in C_i : \phi(v) = f\}\Big| \;\leq\; \alpha_i \cdot \Big|\{v \in C : \phi(v) = f\}\Big|, \quad \forall f \in S, \forall i \in [\ell], \tag{RD}$$

$$\Big|\{v \in C_i : \phi(v) = f\}\Big| \;\geq\; \beta_i \cdot \Big|\{v \in C : \phi(v) = f\}\Big|, \quad \forall f \in S, \forall i \in [\ell], \tag{MP}$$

The assignment ϕ defines a cluster $\{v : \phi(v) = f\}$ around every open facility $f \in S$. As explained in the Introduction, eq. (RD) is the *restricted dominance* property which upper bounds the ratio of any group's participation in a cluster, and eq. (MP) is the *minority protection* property which lower bounds this ratio to protect against under-representation. Due to these fairness constraints, we can no longer assume $\phi(v)$ is the nearest open facility in S to v. Indeed, we use the tuple (S, ϕ) to denote a fair-clustering solution.

We use $\mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$ to denote the optimal value of any instance $\mathcal{I}$ of the FAIR (k,p)-CLUSTERING problem. Since $\mathcal{I}$ is also an instance of the vanilla problem, and since every fair solution is also a vanilla solution (but not necessarily vice versa) we get $\mathrm{OPT}_{\mathsf{vnll}}(\mathcal{I}) \leq \mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$ for any $\mathcal{I}$.

A fair clustering solution (S, ϕ) has λ-*additive* violation, if the eq. (RD) and eq. (MP) constraints are satisfied upto $\pm\lambda$-violation. More precisely, for any $f \in S$ and for any group $i \in [\ell]$, we have

$$\beta_i \cdot \Big|\{v \in C : \phi(v) = f\}\Big| - \lambda \leq \Big|\{v \in C_i : \phi(v) = f\}\Big| \leq \alpha_i \cdot \Big|\{v \in C : \phi(v) = f\}\Big| + \lambda \qquad \text{(V)}$$

Our main result is the following.

Theorem 1. *Given a ρ-approximate algorithm $\mathcal{A}$ for the* VANILLA (k,p)-CLUSTERING *problem, we can return a $(\rho+2)$-approximate solution (S, ϕ) with $(4\Delta+3)$-additive violation for the* FAIR (k,p)-CLUSTERING *problem.*

In particular, we get $O(1)$-factor approximations to the FAIR (k,p)-CLUSTERING problem with $O(\Delta)$ *additive* violation, for any ℓ_p norm[2]. Furthermore, for the important special case of $\Delta = 1$, our additive violation is at most $+3$.

Our technique also implies algorithms for the *lower bounded k-clustering problem*. In this, there is no fairness constraint; rather, the constraint is that if a facility is opened, *at least L* clients must be assigned to it. In this problem also, a client need not be assigned to its nearest open facility. The problem has been studied in the facility location version and (large) $O(1)$-factor algorithms are known. We can easily get the following.

Theorem 2. *Given a ρ-approximate algorithm $\mathcal{A}$ for the* VANILLA (k,p)-CLUSTERING *problem that runs in time T, there is a $(\rho+2)$-approximation algorithm for the* LB-(k,p)-CLUSTERING *problem that runs in time $O(T + 2^k \cdot \mathrm{poly}(n))$.*

In particular, armed with the recent result of Cohen-Addad et al. [28], implies a 3.736-factor approximation algorithm for the lower bounded k-median problem in $k^{O(k)}\mathrm{poly}(n)$ time.

3 Algorithm for the FAIR (k,p)-CLUSTERING problem

Our algorithm is a simple two step procedure. First, we solve the VANILLA (k,p)-CLUSTERING problem using some algorithm $\mathcal{A}$, and fix the centers S opened by $\mathcal{A}$. Then, we solve a *fair reassignment problem*, called FAIR p-ASSIGNMENT problem, on the same set of facilities to get assignment ϕ. We return (S, ϕ) as our fair solution.

Definition 2 (FAIR p-ASSIGNMENT Problem)**.** In this problem, we are given the original set of clients C and a set $S \subseteq F$ with $|S| = k$. The objective is to find the assignment $\phi : C \to S$ such that (a) the constraints eq. (RD) and eq. (MP) are satisfied, and (b) $\mathcal{L}_p(S; \phi)$ is minimized among all such satisfying assignments.

[2] We cannot find an explicit reference for a vanilla $O(1)$-approximation for general ℓ_p norms, but it is not too hard to see that many k-median algorithms such as those of [23] and [44] imply $O(1)$-approximations for any norm. The only explicit mention of the p-norm clustering we could find was the paper [39]; this shows that *local search* gives a $\Theta(p)$ approximation.

Given an instance $\mathcal{J}$ of the FAIR p-ASSIGNMENT problem, we let $\mathrm{OPT}_{\mathsf{asgn}}(\mathcal{J})$ denote its optimum value. Clearly, given any instance $\mathcal{I}$ of the FAIR (k,p)-CLUSTERING problem, if S^* is the optimal subset for $\mathcal{I}$ and $\mathcal{J}$ is the instance of FAIR p-ASSIGNMENT defined by S^*, then $\mathrm{OPT}_{\mathsf{fair}}(\mathcal{I}) = \mathrm{OPT}_{\mathsf{asgn}}(\mathcal{J})$. A λ-violating algorithm for the FAIR p-ASSIGNMENT problem is allowed to incur λ-additive violation to the fairness constraints.

We present our algorithmic template for the FAIR (k,p)-CLUSTERING problem in Algorithm 1. This template uses the FAIRASSIGNMENT procedure (Algorithm 2) as a subroutine.

Algorithm 1 Algorithm for the FAIR (k,p)-CLUSTERING problem

```
procedure FAIRCLUSTERING((𝒳 = F ∪ C, d), C = ∪_{i=1}^ℓ C_i, α⃗, β⃗ ∈ [0,1]^ℓ)
    solve the VANILLA (k,p)-CLUSTERING problem on (𝒳, d)
    let (S, φ) be the solution
    φ̂ = FAIRASSIGNMENT ((𝒳, d), S, C = ∪_{i=1}^ℓ C_i, α⃗, β⃗) ( Algorithm 2)
    return (S, φ̂)
```

3.1 Reducing FAIR (k,p)-CLUSTERING to FAIR p-ASSIGNMENT

In this section we present a simple reduction from the FAIR (k,p)-CLUSTERING problem to the FAIR p-ASSIGNMENT problem that uses a VANILLA (k,p)-CLUSTERING solver as a black-box.

Theorem 3. *Given a ρ-approximate algorithm $\mathcal{A}$ for the* VANILLA (k,p)-CLUSTERING *problem and a λ-violating algorithm $\mathcal{B}$ for the* FAIR p-ASSIGNMENT *problem, there is a $(\rho+2)$- approximation algorithm for the* FAIR (k,p)-CLUSTERING *problem with λ-additive violation.*

Proof. Given instance $\mathcal{I}$ of the FAIR (k,p)-CLUSTERING problem, we run $\mathcal{A}$ on $\mathcal{I}$ to get a (not-necessarily fair) solution (S,ϕ). We are guaranteed $\mathcal{L}_p(S;\phi) \le \rho \cdot \mathrm{OPT}_{\mathsf{vnll}}(\mathcal{I}) \le \rho \cdot \mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$. Let $\mathcal{J}$ be the instance of FAIR p-ASSIGNMENT obtained by taking S as the set of facilities. We run algorithm $\mathcal{B}$ on $\mathcal{J}$ to get a λ-violating solution $\hat{\phi}$. We return $(S,\hat{\phi})$.

By definition of λ-violating solutions, we get that $(S,\hat{\phi})$ satisfies eq. (V) and that $\mathcal{L}_p(S,\hat{\phi}) \le \mathrm{OPT}_{\mathsf{asgn}}(\mathcal{J})$. The proof of the theorem follows from the lemma below. □

Lemma 4. $\mathrm{OPT}_{\mathsf{asgn}}(\mathcal{J}) \le (\rho+2)\cdot \mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$.

Proof. Suppose the optimal solution of $\mathcal{I}$ is (S^*,ϕ^*) with $\mathcal{L}_p(S^*;\phi^*) = \mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$. Recall (S,ϕ) is the solution returned by the ρ-approximate algorithm $\mathcal{A}$. We describe the existence of an assignment $\phi' : C \to S$ such that ϕ' satisfies eq. (RD) and eq. (MP), and $\mathcal{L}_p(S;\phi') \le (\rho+2)\cdot \mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$. Since ϕ' is a feasible solution of $\mathcal{J}$, the lemma follows. For every $f^* \in S^*$, define $\mathsf{nrst}(f^*) := \arg\min_{f\in S} d(f,f^*)$ be the closest facility in S to f^*. For every client $v \in C$, define $\phi'(v) := \mathsf{nrst}(\phi^*(v))$ The following two claims prove the lemma. □

Claim 5. ϕ' satisfies eq. (RD) and eq. (MP)

Proof. For any facility $f^* \in S^*$, let $C(f^*) := \{v : \phi^*(v) = f^*\}$. The $C(f^*)$'s partition C. For any $i \in [\ell]$, let $C_i(f^*) := C(f^*) \cap C_i$. Since $(S^*;\phi^*)$ is a feasible solution satisfying the fairness constraints, we get that for every $f^* \in S^*$ and for every $i \in [\ell]$, $\beta_i \le \frac{|C_i(f^*)|}{|C(f^*)|} \le \alpha_i$.

For any facility $f \in S$, let $N(f) := \{f^* \in S^* : \mathsf{nrst}(f^*) = f\}$ be all the facilities in S^* for which f is the nearest facility. Note that the clients $\{v \in C : \phi'(v) = f\}$ are precisely $\dot\cup_{f^*\in N(f)} C(f^*)$. Similarly, for any $i \in [\ell]$, we have $\{v \in C_i : \phi'(v) = f\}$ is precisely $\dot\cup_{f^* \in N(f)} C_i(f^*)$. Therefore, $\frac{|\{v \in C_i : \phi'(v) = f\}|}{|\{v \in C : \phi'(v) = f\}|} = \frac{\sum_{f^* \in N(f)} |C_i(f^*)|}{\sum_{f^* \in N(f)} |C(f^*)|} \in [\beta_i, \alpha_i]$ since the second summation is between $\min_{f^* \in N(f)} |C_i(f^*)|/|C(f^*)|$ and $\max_{f^* \in N(f)} |C_i(f^*)|/|C(f^*)|$, and both these are in $[\beta_i, \alpha_i]$. ☐

Claim 6. $\mathcal{L}_p(S; \phi') \le (\rho + 2)\,\mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$.

Proof. Fix a client $v \in C$. For the sake of brevity, let: $f = \phi(v)$, $f' = \phi'(v)$, and $f^* = \phi^*(v)$. We have

$$d(v, f') = d(v, \mathsf{nrst}(f^*)) \le d(v, f^*) + d(f^*, \mathsf{nrst}(f^*)) \le d(v, f^*) + d(f^*, f) \le 2d(v, f^*) + d(v, f)$$

The first and third follows from triangle inequality while the second follows from the definition of nrst. Therefore, if we define the assignment cost vectors corresponding to ϕ, ϕ', and ϕ^* as $\vec{d} = \{d(v, \phi) : v \in C\}$, $\vec{d'} = \{d(v, \phi') : v \in C\}$, and $\vec{d^*} = \{d(v, \phi^*) : v \in C\}$ respectively, the above equation implies $\vec{d'} \le 2\vec{d} + \vec{d^*}$. Now note that the $\mathcal{L}_p$ is a monotone norm on these vectors, and therefore,

$$\mathcal{L}_p(S; \phi') = \mathcal{L}_p(\vec{d'}) \le 2\mathcal{L}_p(\vec{d}) + \mathcal{L}_p(\vec{d^*}) = 2\mathcal{L}_p(S^*; \phi^*) + \mathcal{L}_p(S; \phi)$$

The proof is complete by noting $\mathcal{L}_p(S^*; \phi^*) = \mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$ and $\mathcal{L}_p(S; \phi) \le \rho \cdot \mathrm{OPT}_{\mathsf{fair}}(\mathcal{I})$. ☐

3.2 Algorithm for the FAIR p-ASSIGNMENT problem

To complete the proof of Theorem 1, we need to give an algorithm for the FAIR p-ASSIGNMENT problem. We present this in Algorithm 2. The following theorem then establishes our main result.

Theorem 7. *There exists a* $(4\Delta + 3)$*-violating algorithm for the* FAIR p-ASSIGNMENT *problem.*

Proof. Fix an instance $\mathcal{J}$ of the problem. We start by writing a natural LP-relaxation[3].

$$\mathsf{LP} := \min \sum_{v \in C, f \in S} d(v, f)^p x_{v,f} \qquad x_{v,f} \in [0, 1], \;\; \forall v \in C, f \in S \tag{LP}$$

$$\beta_i \sum_{v \in C} x_{v,f} \;\le\; \sum_{v \in C_i} x_{v,f} \;\le\; \alpha_i \sum_{v \in C} x_{v,f} \qquad \forall f \in S, \forall i \in [\ell] \tag{1a}$$

$$\sum_{f \in S} x_{v,f} \;=\; 1 \qquad \forall v \in C \tag{1b}$$

Claim 8. $\mathsf{LP} \le \mathrm{OPT}_{\mathsf{asgn}}(\mathcal{J})^p$.

Proof. Given an optimal solution ϕ^* of $\mathcal{J}$, set $x_{v,f} = 1$ iff $\phi^*(v) = f$. This trivially satisfies the fairness conditions. Observe $\mathcal{L}_p(S; \phi^*)^p$ is precisely the objective cost. ☐

Let $x^\star$ be an optimum solution to the above LP. Note that $x^\star$ could have many coordinates *fractional*. In Algorithm 2, we *iteratively round* $x^\star$ to an *integral* solution with the same or better value, but which violates the fairness constraints by at most $4\Delta + 3$. Our algorithm effectively simulates an algorithm for *minimum degree-bounded matroid basis* problem (MBDMB henceforth) due to Király et al. [49]. In this problem one is given a matroid $M = (X, \mathcal{I})$, costs on elements in X, a hypergraph $H = (X, \mathcal{E})$, and functions $f : \mathcal{E} \to \mathbb{R}$ and $g : \mathcal{E} \to \mathbb{R}$ such that $f(e) \le g(e)$ for all $e \in \mathcal{E}$. The objective is to find the minimum cost basis $B \subseteq X$ such that for all $e \in \mathcal{E}$, $f(e) \le |B \cap e| \le g(e)$. We state the main result in Király et al [49] below.

[3]This makes sense only for finite p. See Remark 1

Theorem 9 (Paraphrasing of Theorem 1 in [49])**.** *There exists a polynomial time algorithm that outputs a basis B of cost at most* OPT*, such that $f(e) - 2\Delta_H + 1 \le |B \cap e| \le g(e) + 2\Delta_H - 1$ for each edge $e \in \mathcal{E}$ of the hypergraph, where $\Delta_H = max_{v \in X}|\{e \in E_H : v \in e\}|$ is the maximum degree of a vertex in the hypergraph H, and* OPT *is the cost of the natural LP relaxation.*

To complete the proof of the main theorem, we first construct an instance of the MBDMB problem using $x^\star$. Then we appeal to Theorem 9 to argue about the quality of our algorithm.

Let E be the set of (v, f) pairs with $x^\star_{v,f} > 0$. For a point $v \in C$, let E_v denote the set of edges in E incident on v. Define $\mathcal{F} := \{F \subseteq E : |F \cap E_v| \le 1 \ \ \forall v \in C\}$ to be collection of edges which "hit" every client at most once. The pair $M = (E, \mathcal{F})$ is a well known combinatorial object called a (partition) *matroid*. For each element (v, f) of this matroid M, we denotes its cost to be $c(v, f) := d(v, f)^p$.

Next we define a *hypergraph* $H = (E, \mathcal{E})$. For each $f \in S$ and $i \in [\ell]$, let $E_{f,i} \subseteq E$ consisting of pairs $(v, f) \in E$ for $v \in C_i$. Let $E_f := \cup_{i=1}^{\ell} E_{f,i}$. Each of these $E_{f,i}$'s and E_f's are added to the collection of hyperedges $\mathcal{E}$. Next, let $T_f := \sum_{v \in C} x^\star_{v,f}$ be the total fractional assignment on f. Similarly, for all $i \in [\ell]$, define $T_{f,i} := \sum_{v \in C_i} x^\star_{v,f}$. Note that, both T_f and $T_{f,i}$ can be fractional. For every $e \in E_{f,i}$, we define $f(e) := \lfloor T_{f,i} \rfloor$ and $g(e) = \lceil T_{f,i} \rceil$. For each $e \in E_f$, we denote $f(e) = \lfloor T_f \rfloor$ and $g(e) = \lceil T_f \rceil$. This completes the construction of the MBDMB instance.

Now we can apply Theorem 9 to obtain a basis B of matroid M with the properties mentioned. Note that for our hypergraph $\Delta_H \le \Delta + 1$ where Δ is the maximum number of groups a client can be in. This is because every pair (v, f) belongs to E_f and $E_{f,i}$'s for all C_i's containing v. Also note that any basis corresponds to an assignment $\phi : C \to S$ of all clients. Furthermore, the cost of the basis is precisely $\mathcal{L}_p(S; \phi)^p$. Since this cost is $\le \mathsf{LP} \le \mathrm{OPT}_{\mathsf{fair}}(\mathcal{J})^p$, we get that $\mathcal{L}_p(S; \phi) \le \mathrm{OPT}_{\mathsf{fair}}(\mathcal{J})$. We now need to argue about the violation.

Fix a server f and a client group C_i. Let $\overline{T}_f$ and $\overline{T}_{f,i}$ denote the number of clients assigned to f and the number of clients from C_i that are assigned to f respectively (by the integral assignment). Then, by Theorem 9, $\lfloor T_f \rfloor - 2\Delta - 1 \le \overline{T}_f \le \lceil T_f \rceil + 2\Delta + 1$ and $\lfloor T_{f,i} \rfloor - 2\Delta - 1 \le \overline{T}_{f,i} \le \lceil T_{f,i} \rceil + 2\Delta + 1$ (using $\Delta_H \le \Delta + 1$). Now consider eq. (RD). Since, $T_{f,i} \le \alpha_i T_f$ (as the LP solution is feasible),

$$\overline{T}_{f,i} \le \lceil \alpha_i T_f \rceil + 2\Delta + 1 \le \alpha_i \lfloor T_f \rfloor + 2\Delta + 2 \le \alpha_i(\overline{T}_f + 2\Delta + 1) + 2\Delta + 2 \le \alpha_i \overline{T}_f + (4\Delta + 3)\,,$$

where the second and last inequality follows as $\alpha_i \le 1$. We can similarly argue about eq. (MP). This completes the proof of Theorem 7. □

However, rather than constructing an MBDMB instance explicitly, we write a natural LP-relaxation more suitable to the task — this is given in eq. (2). For the sake of completeness, we give the details of our algorithm in algorithm 2.

$$\mathsf{LP2} := \min \sum_{v \in C, f \in S} d(v,f)^p x_{v,f} \qquad x_{v,f} \in [0,1], \ \ \forall v \in C, f \in S \tag{2a}$$

$$\lfloor T_f \rfloor \ \le \ \sum_{v \in C} x_{v,f} \ \le \ \lceil T_f \rceil \qquad \forall f \in S, \forall i \in [\ell] \tag{2b}$$

$$\lfloor T_{f,i} \rfloor \ \le \ \sum_{v \in C_i} x_{v,f} \ \le \ \lceil T_{f,i} \rceil \qquad \forall f \in S, \forall i \in [\ell] \tag{2c}$$

$$\sum_{f \in S} x_{v,f} \ = \ 1 \qquad \forall v \in C \tag{2d}$$

Algorithm 2 Algorithm for the FAIR p-ASSIGNMENT problem

procedure FAIRASSIGNMENT($(\mathcal{X}, d)$, S, $C = \cup_{i=1}^{\ell} C_i$, $\vec{\alpha}, \vec{\beta} \in [0,1]^{\ell}$)
    $\hat{\phi}(v) = \emptyset$ for all $v \in C$
    solve the LP given in eq. (1), let $x^\star$ be an optimal solution
    for each $x^\star_{v,f} = 1$, set $\hat{\phi}(v) = f$ and remove v from C (and relevant C_is).
    let $T_f := \sum_{v \in C} x^\star_{v,f}$ for all $f \in S$
    let $T_{f,i} := \sum_{v \in C_i} x^\star_{v,f}$ for all $i \in [\ell]$ and $f \in S$
    construct LP2 as given in eq. (2), only with variables $x_{v,f}$ such that $x^\star_{v,f} > 0$
    while there exists a $v \in C$ such that $\hat{\phi}(v) = \emptyset$ **do**
        solve LP2, let $x^\star$ be an optimal solution
        for each $x^\star_{v,f} = 0$, delete the variable $x^\star_{v,f}$ from LP2
        for each $x^\star_{v,f} = 1$, set $\hat{\phi}(v) = f$ and remove v from C (and relevant C_is). Reduce T_f and relevant $T_{f,i}$'s by 1.
        for every $i \in [\ell]$ and $f \in S$, if $|x^\star_{v,f} : 0 < x^\star_{v,f} < 1,\ v \in C_i| \leq 2(\Delta + 1)$ remove the respective constraint in eq. (2c)
        for every $f \in S$, if $|x^\star_{v,f} : 0 < x^\star_{v,f} < 1,\ v \in C| \leq 2(\Delta + 1)$ remove the respective constraint in eq. (2b)

Remark 1. For the case of $p = \infty$, the objective function of eq. (LP) doesn't make sense. Instead, one proceeds as follows. We begin with a guess G of $\mathrm{OPT}_{\mathsf{asgn}}(\mathcal{J})$; we set $x_{v,f} = 0$ for all pairs with $d(v,f) > G$. We then check if eqs. (1a) and (1b) have a feasible solution. If they do not, then our guess G is infeasible (too small). If they do, then the proof given above returns an assignment which violates eqs. (RD) and (MP) by additive $4\Delta + 3$, and satisfies $d(v, \phi(v)) \leq G$ for all $v \in C$.

Remark 2. When $\Delta = 1$, that is, the C_i's are disjoint, we can get an improved $+3$ additive violation (instead of $+7$). Instead of using Theorem 9, we use the generalized assignment problem (GAP) rounding technique by Shmoys and Tardos [62] to achieve this.

Remark 3. Is having a bicriteria approximation necessary? We do not know. The nub is the FAIR p-ASSIGNMENT problem. It is not hard to show that deciding whether a λ-violating solution exists with $\lambda = 0$ under the given definition is NP-hard. [4] However, an algorithm with $\lambda = 0$ and cost within a constant factor of $\mathrm{OPT}_{\mathsf{asgn}}(\mathcal{J})$ is not ruled out. This is an interesting open question.

4 Lower-bounded clustering

In this section we show a simple application of our technique which solves the lower bounded clustering problem. The problem arises when, for example, one wants to ensure anonymity [4] and is called "private clustering" in [60]. For the $p = \infty$ norm, that is the *lower bounded k-center* problem, there is a 3-approximation known [4] for the problem. For the $p = 1$ norm, that is the *lower bounded k-median problem*, there are $O(1)$-approximation algorithms[5] [63, 8] although the constants are large. In contrast, we show simple algorithms with much better constants in $O(2^k \mathrm{poly}(n))$ time.

[4] A simple reduction from the 3D-matching problem.

[5] Actually, the papers of [63, 8] consider the facility location version without any constraints on the number of facilities. A later paper by Ahmadian and Swamy [9] mentions that these algorithms imply $O(1)$-approximations for the k-median version. The constant is not specified.

Definition 3 (LB-(k,p)-CLUSTERING)**.** The input is a VANILLA (k,p)-CLUSTERING instance, and an integer $L \in [|C|]$. The objective is to open a set of facilities $S \in \mathcal{F}$ with $|S| \leq k$, and find an assignment function $\phi : C \to S$ of clients to the opened facilities so that (a) $\mathcal{L}_p(S;\phi)$ is minimized, and (b) for every $f \in S$, we have $|\{v \in C : \phi(v) = f\}| \geq L$.

Theorem 2. *Given a ρ-approximate algorithm $\mathcal{A}$ for the* VANILLA (k,p)-CLUSTERING *problem that runs in time T, there is a $(\rho+2)$-approximation algorithm for the* LB-(k,p)-CLUSTERING *problem that runs in time $O(T + 2^k \cdot \mathrm{poly}(n))$.*

Using the best known polynomial time algorithm for the k-median problem due to Byrka et al. [16] and best known FPT-algorithm due to Cohen-Addad et al. [28], we get the following corollary.

Theorem 10. *There is a 4.676-factor approximation algorithm for the lower bounded k-median running in $O(2^k \cdot \mathrm{poly}(n))$ time. There is a 3.736-factor approximation algorithm for lower bounded k-median running in time $k^{O(k)}\mathrm{poly}(n)$ time.*

Remark 4. As in the case of fair clustering, Theorem 2 holds even when there are more general constraints on the centers. Therefore, for instance, in $O(2^k\mathrm{poly}(n))$ time, we can get a 34-approximation for the lower bounded knapsack median problem due to the knapsack median result [64], and a 5-approximation for the lower bounded center problem even when the total weight of the centers is at most a bound and the set of centers need be an independent set of a matroid [21].

Proof of Theorem 2. The proof is nearly identical to that of Theorem 3. Given an instance $\mathcal{I}$ of the LB-(k,p)-CLUSTERING problem, we first run algorithm $\mathcal{A}$ to get (S,ϕ) with the property $\mathcal{L}_p(S;\phi) \leq \rho \cdot \mathrm{OPT}_{\mathsf{vnll}}(\mathcal{I}) \leq \rho \cdot \mathrm{OPT}_{\mathsf{lbnd}}(\mathcal{I})$.

Now we construct 2^k instances of the b-matching problem with lower bounds. For *every* subset $T \subseteq S$ we construct a complete bipartite graph on $(C \cup T)$ with cost of edge $c(v,f) := d(v,f)^p$ for all $v \in C, f \in T$. There is a lower bound of 1 on every $v \in C$ and L on every $f \in T$. We find a minimum cost matching satisfying these. Given a matching M, the assignment is the natural one: $\phi(v) = f$ if $(v,f) \in M$. We return the assignment $\hat{\phi}$ of minimum cost among these 2^k possibilities. Note that $\mathcal{L}_p(T;\hat{\phi})$ equals the $(1/p)$th power of the cost of this matching.

To analyze the above algorithm, we need to show the existence of some subset $T \subseteq S$ such that the minimum cost lower bounded matching M has $c(M)^{1/p} \leq (\rho+2)\,\mathrm{OPT}_{\mathsf{lbnd}}(\mathcal{I})$. The proof is very similar to the proof of Lemma 4. Suppose (S^*,ρ^*) is the optimal solution for $\mathcal{I}$ of cost $\mathrm{OPT}_{\mathsf{lbnd}}(\mathcal{I})$. For any $f^* \in S^*$, define $\mathsf{nrst}(f^*)$ to be its closest facility in S. Let $T := \{f \in S : \exists f^* \in S^* \text{s.t. } f = \mathsf{nrst}(f^*)\}$; by definition $T \subseteq S$. Define the matching where for each $v \in C$ we match v to $\mathsf{nrst}(\phi^*(v)) \in T$. As in the proof of Lemma 4, one can show that $d(v, \mathsf{nrst}(\phi^*(v))) \leq 2d(v,\phi^*(v)) + d(v,\phi(v))$. Thus, the $1/p$th power of the cost of the matching is at most $(\rho+2)\cdot\mathrm{OPT}_{\mathsf{lbnd}}(\mathcal{I})$. Furthermore, for every $f \in T$, the number of clients assigned to it is at least the number of clients assigned to an f^* with $\mathsf{nrst}(f^*) = f$. This is $\geq L$. □

Remark 5. Finally, we mention that the above prove easily generalizes for the notion of *strong privacy* proposed by Rösner and Schmidt [60]. In this, the client set is *partitioned* into groups $C_1,\ldots,C_\ell$, and the goal is to assign clients to open facilities such that for every facility the number of clients from a group C_i is at least L_i. We can generalize Theorem 2 to get a $(\rho+2)$-approximation algorithm for this problem running in $O(2^k\mathrm{poly}(n))$ time whenever there was a ρ-approximation possible for the vanilla clustering version.

5 Experiments

In this section, we perform empirical evaluation of our algorithm. Based on our experiments, we report five key findings:

- Vanilla clustering algorithms are quite *unfair* even when measured against relaxed settings of α and β. In contrast, our algorithm's additive violation is almost always less than 3, even with $\Delta = 2$, across a wide range of parameters (see section 5.1).
- The cost of our fair clustering is at most 15% more than (unfair) vanilla cost for $k \leq 10$ as in fig. 3. In fact, we see (in fig. 4) that our algorithm's cost is *very close* to the *absolute best* fair clustering that allows additive violations! Furthermore, our results for k-median significantly improve over the costs reported in Chierichetti et al. [25] and Backurs et al. [12] (see Table 3).
- For the case of overlapping protected groups ($\Delta > 1$), enforcing fairness with respect to one sensitive attribute (say gender) can lead to unfairness with respect to another (say race). This empirical evidence stresses the importance of considering $\Delta > 1$ (see fig. 2 in Section 5.3).
- In section 5.4, we provide experiments to gauge the running time of our algorithm in practice on large datasets. Even though the focus of this paper is not optimizing the running time, we observe that our algorithm for the k-means objective finds a fair solution for the `census1990` dataset with 500K points and 13 features in less than 30 minutes (see Table 4).
- Finally, we study how the cost of our fair clustering algorithm changes with the strictness of the fairness conditions. This enables the user to figure out the trade-offs between fairness and utility and make an informed decision about which threshold to choose (see Section 5.5).

Settings. We implement our algorithm in Python 3.6 and run all our experiments on a Macbook Air with a 1.8 GHz Intel Core i5 Processor and 8 GB 1600 MHz DDR3 memory. We use CPLEX[43] for solving LP's. Our codes are available on GitHub[6] for public use.

Datasets. We use four datasets from the UCI repository [33]: [7] **(1)** `bank` [65] with 4,521 points, corresponding to phone calls from a marketing campaign by a Portuguese banking institution. **(2)** `census` [53] with 32,561 points, representing information about individuals extracted from the 1994 US `census` . **(3)** `creditcard` [42] with 30,000 points, related to information on credit card holders from a certain credit card in Taiwan. **(4)** `census1990` [57] with 2,458,285 points, taken from the 1990 US Census, which we use for run time analysis. For each of the datasets, we select a set of numerical attributes to represent the records in the Euclidean space. We also choose two sensitive attributes for each dataset (e.g. sex and race for `census`) and create protected groups based on their values. Table 1 contains a more detailed description of the datasets and our features.

Measurements. For any clustering, we mainly focus on two metrics. One is the *cost of fairness*, that is, the ratio of the objective values of the fair clustering over the vanilla clustering. The other is *balance*, the measure of unfairness. To define balance, we generalize the notion found by Chierichetti et al. [25], We define two intermediate values r_i, the representation of group i in the dataset and $r_i(f)$, the representation of group i in cluster f as $r_i := |C_i|/|C|$ and $r_i(f) := |C_i(f)|/|C(f)|$. Using these two values, balance is defined as $\text{balance}(f) := \min\{r_i/r_i(f), r_i(f)/r_i\}\ \ \forall i \in [\ell]$. Although in theory the values of α, β for a given group i can be set arbitrarily, in practice they are best set with respect to r_i, the ratio of the group in

[6] https://github.com/nicolasjulioflores/fair_algorithms_for_clustering
[7] https://archive.ics.uci.edu/ml/datasets/

Table 1: For each dataset, the coordinates are the numeric attributes used to determined the position of each record in the Euclidean space. The sensitive attributes determines protected groups.

Dataset	Coordinates	Sensitive attributes	Protected groups
`bank`	age, balance, duration	marital	married, single, divorced
		default	yes, no
`census`	age, education-num, final-weight, capital-gain, hours-per-week	sex	female, male
		race	Amer-ind, asian-pac-isl, black, other, white
`creditcard`	age, bill-amt 1 — 6, limit-bal, pay-amt 1 — 6	marriage	married, single, other, null
		education	7 groups
`census1990`	dAncstry1, dAncstry2, iAvail, iCitizen, iClass, dDepart, iFertil, iDisabl1, iDisabl2, iEnglish, iFeb55, dHispanic, dHour89	dAge	8 groups
		iSex	female, male

the dataset. Furthermore, to reduce the degrees of freedom, we parameterize β and α by a single variable δ such that $\beta_i = r_i(1-\delta)$ and $\alpha_i = r_i/(1-\delta)$. Thus, we can interpret δ as how loose our fairness condition is. This is because $\delta = 0$ corresponds to each group in each cluster having exactly the same ratio as that group in the dataset, and $\delta = 1$ corresponds to no fairness constraints at all. For all of the experiments, we set $\delta = 0.2$ (corresponding to the common interpretation of the 80%-rule of DI doctrine), and use $\Delta = 2$, unless otherwise specified.

Algorithms. For vanilla k-center, we use a 2-approx. algorithm due to Gonzalez [38]. For vanilla k-median, we use the single-swap 5-approx. algorithm by Arya et al. [13], augment it with the D-sampling procedure by [11] for initial center section, and take the best out of 5 trials. For k-means, we use the k-means++ implementation of [58].

5.1 Fairness comparison with vanilla clustering

In fig. 1 we motivate our discussion of fairness by demonstrating the unfairness of vanilla clustering and fairness of our algorithm. On the x-axis, we compare three solutions: (1) our algorithm (labelled "ALG"), (2) fractional solution to the FAIR p-ASSIGNMENT LP in Equation (1) (labelled "Partial"), and (3) vanilla k-means (labelled "VC"). Below these labels, we record the *cost of fairness*. We set $\delta = 0.2$ and $k = 4$. Along the y axis, we plot the balance metric defined above for the three largest clusters for each of these clustering. The dotted line at 0.8 is the goal balance for $\delta = 0.2$. The lowest balance for any cluster for our algorithm is 0.75 (for `census`), whereas vanilla can be as bad as 0 (for `bank`); "partial" is, of course, always fair (at least 0.8). We observe that the maximum additive violation of our algorithm is only 3 (much better than our theoretical bound of $4\Delta+3$)), for a large range of values of δ and k, whereas vanilla k-means can be unfair by quite a large margin. (see fig. 2 below and Table 2).

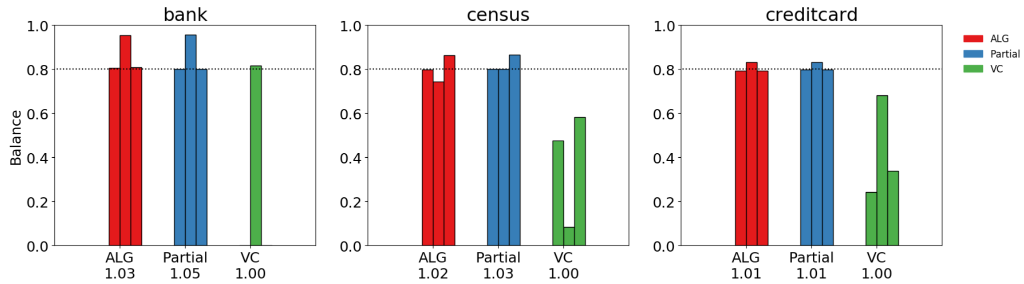

Figure 1: Comparison of our algorithm (ALG) versus vanilla clustering (VC) in terms of balance for the k-means objective.

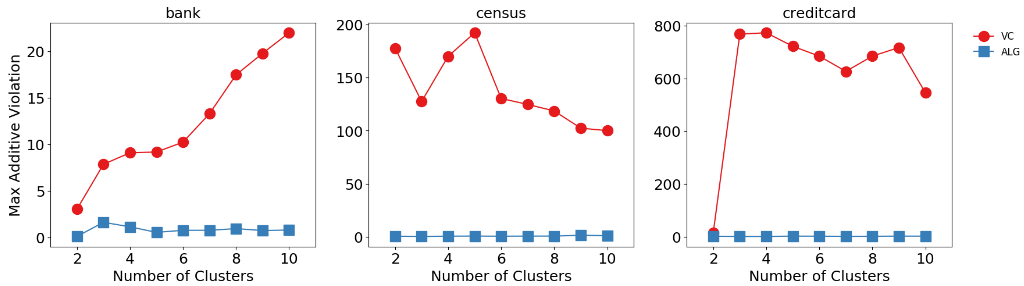

Figure 2: Comparison of the maximum additive violation (for $\delta = 0.2$ and $\Delta = 2$) over all clusters and all groups between our algorithm (ALG) and vanilla (VC), using the k-means objective.

Table 2: The maximum additive violation across a range of δ of our algorithm compared to vanilla k-means. For each δ, we take maximum over k, for $k \in [2, 10]$ on all datasets.

δ	0.01	0.05	0.1	0.2	0.3	0.4	0.5	Vanilla ($\delta = 0.2$)
`bank`	1.45	1.17	1.39	**1.54**	1.19	1.15	1.03	**21.99**
`census`	1.44	1.53	**1.89**	1.08	1.18	0.97	1.03	**773.19**
`creditcard`	**3.02**	2.32	2.11	2.29	2.03	1.63	1.03	**192.01**

5.2 Cost analysis

We evaluate the cost of our algorithm for k-means objective with respect to the vanilla clustering cost. Figure 3 shows that the cost of our algorithm for $k \leq 10$ is at most 15% more than the vanilla cost for all datasets. Interestingly, for `creditcard` , even though the vanilla solution is extremely unfair as demonstrated earlier, cost of fairness is at most 6% which indicates that the vanilla centers are in the "right place".

Our results in Table 3 confirm that we outperform [25] and [12] in terms of cost. To match [25] and [12], we sub-sample `bank` and `census` to 1000 and 600 respectively, declared only one sensitive attribute for each (i.e. marital for `bank` and sex for `census`), and tune the fairness parameters to enforce a balance of 0.5. The data in table 3 is from [12] which is only available for k-median and $k = 20$.

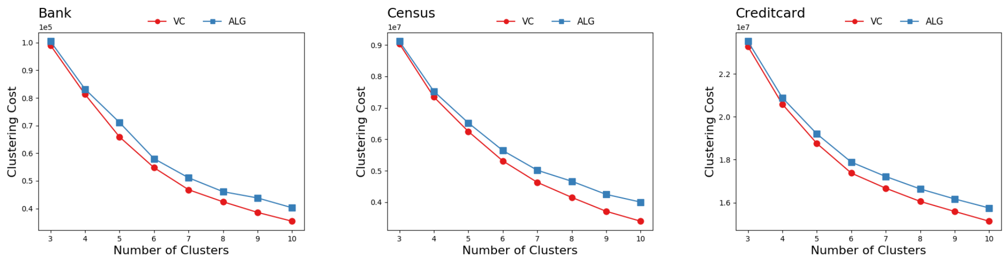

Figure 3: Our algorithm's cost (ALG) versus the vanilla clustering cost (VC) for k-means objective.

Table 3: Comparison of our clustering cost with [12] and [25] for k-median ($k = 20$).

Dataset	Ours	Backurs et al. [12]	Chierichetti et al. [25]
`bank`	$\mathbf{2.43 \times 10^5}$	6.03×10^5	5.55×10^5
`census`	$\mathbf{4.24 \times 10^6}$	24.1×10^6	36.5×10^6

Next, we evaluate the cost of our algorithm for k-means objective with respect to the vanilla clustering cost and the *almost fair LP* cost. The almost fair LP (eq. (3)) is an LP relaxation of FAIR (k, p)-CLUSTERING, with variables for choosing the centers, except that we allow for a λ additive violation in fairness. The cost of this LP is a lower-bound on the cost of *any* fair clustering that violates fairness by at most an additive factor of λ.

$$\mathsf{LP3} := \min \sum_{v \in C, f \in S} d(v,f)^p x_{v,f} \qquad x_{v,f} \in [0,1], \ \ \forall v \in C, f \in S \tag{3a}$$

$$\sum_{f \in S} x_{v,f} = 1 \qquad \forall v \in C \tag{3b}$$

$$x_{v,f} \leq y_f \qquad \forall v \in C, f \in S \tag{3c}$$

$$\sum_{f \in S} y_f \leq k \tag{3d}$$

$$\sum_{v \in C_i} x_{v,f} \leq \alpha_i \sum_{v \in C} x_{v,f} + \lambda \qquad \forall f \in S, \forall i \in [\ell] \tag{3e}$$

$$\sum_{v \in C_i} x_{v,f} \geq \beta_i \sum_{v \in C} x_{v,f} - \lambda \qquad \forall f \in S, \forall i \in [\ell] \tag{3f}$$

In fig. 4 we compare the cost of our algorithm with a lower-bound on the absolute best cost of any clustering that has the same amount of violation as ours. To be more precise, for any dataset we set λ according to the maximum violation of our algorithm reported in table 2 for $\delta = 0.2$ (e.g. λ is 1.54 for `bank` , 1.08 for `census` , and 2.29 for `creditcard`). Then, we solve the almost fair LP for that λ and compare its cost with our algorithm's cost over that dataset.

Since solving the almost fair LP on the whole data is infeasible (in terms of running time), we subsample `bank` , `census` , and `creditcard` to 1000, 600 and 600 points respectively, and report the average costs over 10 trials. Also, we only consider one sensitive attribute, namely marital for `bank` , sex for `census` and education for `creditcard` to further simplify the LP and decrease the running time.

fig. 3 shows that the cost of our algorithm is very close to the almost fair LP cost (at most 15% more). Note that, since the cost of almost fair LP is a lower bound on the cost of FAIR (k, p)-CLUSTERING problem, we conclude that our cost is at most 15% more than optimum in practice, which is much better than the proved $(\rho + 2)$ factor in theorem 1.

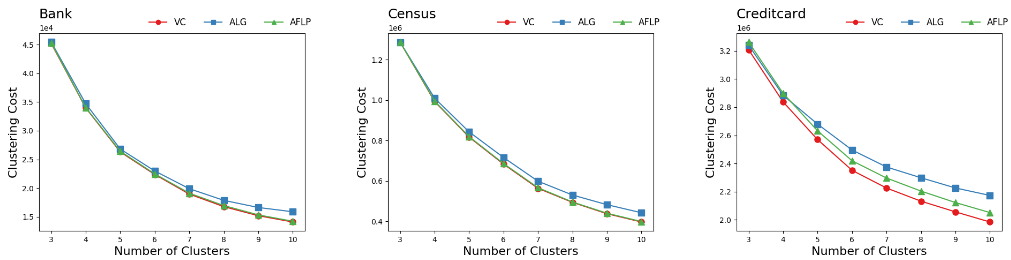

Figure 4: Average costs of vanilla clustering (VC), our algorithm (ALG), and almost fair LP (AFLP), for k-means objective, as a function of k.

5.3 The case of $\Delta > 1$

In this section, we demonstrate the importance of considering $\Delta > 1$ by showing that enforcing fairness with respect to one attribute (say gender) may lead to significant unfairness with respect to another attribute (say race). In Figure 5, we have two plots for each dataset. In each plot, we compare three clustering: (1) Our algorithm with $\Delta = 2$ (labelled "both"); (2) and (3) Our algorithm with $\Delta = 1$ with protected groups defined by the attribute on x-axis label. We set $\delta = 0.2$ and $k = 4$. The clustering objective is k-means. Along y-axis, we measure the *balance* metric for the three largest clusters for each of these clustering. In each plot we only measure the *balance* for the attribute written in bold in the top right corner.

In datasets, such as `bank` , we see that fairness with respect to only the marital attribute leads to a large amount of unfairness in the default attribute. The fairest solution along both attributes is when they are both considered by our algorithm ($\Delta = 2$). Interestingly, there are datasets where fairness by one attribute is all that is needed. On the `census` dataset, fairness by race leads to a fair solution on sex, but fairness by sex leads to large amount of unfairness in race.

Finally, our results strongly suggest that finding a fair solution for two attributes is often only slightly more expensive (in terms of the clustering objective) than finding a fair solution for only one attribute.

5.4 Run time analysis

In this paper, we focus on providing a framework and do not emphasize on run time optimization. Nevertheless, we note that our algorithm for the k-means objective finds a fair solution for the `census1990` dataset with 500K points and 13 features in less than 30 minutes (see Table 4). Even though our approach is based on iterative rounding method, in practice CPLEX solution to LP (eq. (1)) is more than 99% integral for each of our experiments. Hence, we never have to solve more than two or three LP. Also the number of variables in subsequent LPs are significantly small. In contrast, if we attempt to frame LP (eq. (1)) as an integer program instead, the CPLEX solver fails to find a solution in under an hour even with 40K points.

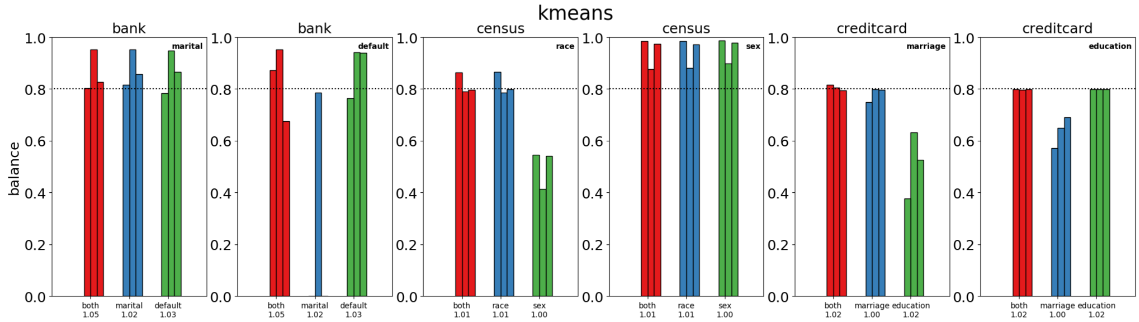

Figure 5: Importance of considering $\Delta > 1$. Below these x labels is the *cost of fairness* ratio. We report the *balance* for the three largest clusters and include the dotted line at 0.8 because we use $\delta = 0.2$.

Table 4: Runtime of our algorithm on subsampled data from `census1990` for k-means ($k = 3$).

Number of sampled points	10K	50K	100K	200K	300K	400K	500K
Time (sec)	4.04	33.35	91.15	248.11	714.73	1202.89	1776.51

5.5 Tuning the fairness parameters

In Figure 6, we demonstrate the ability to tune the strictness of the fairness criteria by manipulating the parameter δ. As δ approaches 1, the ratio between the fair objective and original vanilla objective decreases to 1. This suggests that the fair solution has recapitulated the vanilla clustering because our bounds are lax enough to do so.

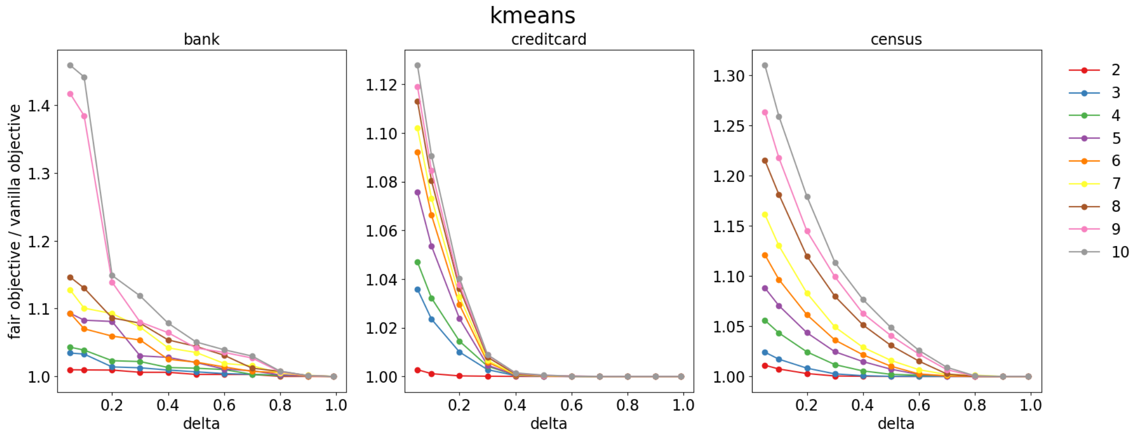

Figure 6: We show the effects of varying δ (x-axis) on our algorithm's fair objective cost over the vanilla cost (y-axis).

References

[1] Supreme Court of the United States. Griggs v. Duke Power Co. 401 U.S. 424, March 8 1971. 2

[2] Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed M. Meesum, and Michal Wlodarczyk. Constant factor FPT approximation for capacitated k-median. *CoRR*, abs/1809.05791, 2018. 4

[3] Charu C Aggarwal and Chandan K Reddy. *Data clustering: algorithms and applications*. CRC press, 2013. 3

[4] Gagan Aggarwal, Tomás Feder, Krishnaram Kenthapadi, Samir Khuller, Rina Panigrahy, Dilys Thomas, and An Zhu. Achieving anonymity via clustering. In *ACM Symposium on Principles of Database Systems*, 2006. 2, 9

[5] Herman Aguinis and Wayne F. Cascio. *Applied Psychology in Human Resource Management (6th Edition)*. Prentice Hall, 2005. 2

[6] Sara Ahmadian, Alessandro Epasto, Ravi Kumar, and Mohammad Mahdian. Clustering without over-representation. In *Proc. 25th Annual SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019. 3

[7] Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and Euclidean k-median by primal-dual algorithms. In *Annual IEEE Symposium on Foundations of Computer Science*, 2017. 4

[8] Sara Ahmadian and Chaitanya Swamy. Improved approximation guarantees for lower-bounded facility location. In *Proceedings, Workshop on Approximation and Online Algorithms (WAOA)*, 2012. 9

[9] Sara Ahmadian and Chaitanya Swamy. Approximation Algorithms for Clustering Problems with Lower Bounds and Outliers. In *International Colloquium on Automata, Languages and Programming*, 2016. 9

[10] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: Theres software used across the country to predict future criminals. and its biased against blacks. *ProPublica*, May 23 2016. 1

[11] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007. 12

[12] Krzysztof Onak Baruch Schieber Ali Vakilian Arturs Backurs, Piotr Indyk and Tal Wagner. Scalable fair clustering. In *Proc. 36th Proceedings, International Conference on Machine Learning (ICML)*, June 2019. 3, 11, 13, 14

[13] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on computing*, 33(3):544–562, 2004. 4, 12

[14] Ioana O. Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R. Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. *CoRR*, abs/1811.10319, 2018. 3

[15] Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster algorithms for the constrained k-means problem. *Theory of Computing Systems*, 62, 2018. 3

[16] Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2014. 4, 10

[17] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010. 3

[18] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, 2019. 3

[19] L Elisa Celis, Lingxiao Huang, and Nisheeth K Vishnoi. Multiwinner voting with fairness constraints. In *IJCAI*, pages 144–151, 2018. 1, 3

[20] L. Elisa Celis, Damian Straszak, and Nisheeth K. Vishnoi. Ranking with Fairness Constraints. In *Proc. 45th International Colloquium on Automata, Languages and Programming*, pages 28:1–28:15, 2018. 1, 2, 3

[21] Deeparnab Chakrabarty and Maryam Negahbani. Generalized Center Problems with Outliers. In *International Colloquium on Automata, Languages and Programming*, 2018. 10

[22] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Annual IEEE Symposium on Foundations of Computer Science*, 1999. 4

[23] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002. 4, 5

[24] Xingyu Chen, Brandon Fain, Charles Lyu, and Kamesh Munagala. Proportionally fair clustering. In *Proc. 36th Proceedings, International Conference on Machine Learning (ICML)*, June 2019. 3

[25] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Proc. 31st Conference on Neural Information Processing Systems*, pages 5029–5037, 2017. 1, 2, 3, 11, 13, 14

[26] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvtiskii. Matroids, matchings, and fairness. In *Proceedings of Machine Learning Research*, volume 89, 2019. 3

[27] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017. 1

[28] Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight fpt approximations for k-median and k-means. *International Colloquium on Automata, Languages and Programming*, 2019. 3, 5, 10

[29] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proc. 23rd Annual SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806. ACM, 2017. 3

[30] Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. In *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2015. 3

[31] Jerry Dischler. Putting machine learning into the hands of every advertiser. https://www.blog.google/technology/ads/machine-learning-hands-advertisers/, July 10 2018. 1

[32] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4, 2018. 1

[33] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. 11

[34] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proc. 3rd Conference on Innovations in Theoretical Computer Science*, pages 214–226. ACM, 2012. 3

[35] The U.S. EEOC. Uniform guidelines on employee selection procedures, March 2 1979. 2

[36] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proc. 21st Annual SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268, 2015. 1, 3

[37] Sorelle A. Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im)possibility of fairness. *CoRR*, abs/1609.07236, 2016. 3

[38] Teofilo F. Gonzalez. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.*, 38:293 – 306, 1985. 4, 12

[39] Anupam Gupta and Kanat Tangwongsan. Simpler analyses of local search algorithms for facility location. *arXiv preprint arXiv:0809.2554*, 2008. 5

[40] Dorit S. Hochbaum and David B. Shmoys. A best possible heuristic for the k-center problem. *Math. Oper. Res.*, 10(2):180–184, 1985. 4

[41] Wen-Lian Hsu and George L Nemhauser. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979. 4

[42] Che-hui Lien I-Cheng Yeh. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 2009. 11

[43] IBM. Ibm ilog cplex 12.9. 2019. 11

[44] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001. 4, 5

[45] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Conference on Neural Information Processing Systems*, pages 325–333, 2016. 1, 3

[46] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50, 2012. 1, 3

[47] Amir E Khandani, Adlar J Kim, and Andrew W Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34, 2010. 1

[48] Samir Khuller and Yoram J Sussmann. The capacitated k-center problem. *SIAM Journal on Discrete Mathematics*, 13(3):403–418, 2000. 4

[49] Tamás Király, Lap Chi Lau, and Mohit Singh. Degree bounded matroids and submodular flows. *Combinatorica*, 32(6):703–720, 2012. 7, 8

[50] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Proc. 8th Conference on Innovations in Theoretical Computer Science*, pages 43:1–43:23, 2017. 3

[51] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *Proc. 36th Proceedings, International Conference on Machine Learning (ICML)*, June 2019. 3

[52] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. In *Proc. 36th Proceedings, International Conference on Machine Learning (ICML)*, June 2019. 3

[53] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Annual SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1996. 11

[54] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016. 4

[55] Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini. k-nn as an implementation of situation testing for discrimination discovery and prevention. In *Proc. 17th Annual SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 502–510, 2011. 3

[56] Rashmi Malhotra and Davinder K Malhotra. Evaluating consumer loans using neural networks. *Omega*, 31, 2003. 1

[57] Christopher Meek, Bo Thiesson, and David Heckerman. The learning-curve sampling method applied to model-based clustering. *Journal of Machine Learning Research*, 2:397, 2002. 11

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011. 12

[59] Claudia Perlich, Brian Dalessandro, Troy Raeder, Ori Stitelman, and Foster Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95, 2014. 1

[60] Clemens Rösner and Melanie Schmidt. Privacy Preserving Clustering with Constraints. In *Proc. 45th International Colloquium on Automata, Languages and Programming*, pages 96:1–96:14, 2018. 1, 2, 3, 9, 10

[61] Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. Fair coresets and streaming algorithms for fair k-means clustering. *arXiv preprint arXiv:1812.10854*, 2018. 3

[62] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical programming*, 62(1-3):461–474, 1993. 9

[63] Zoya Svitkina. Lower-bounded facility location. *ACM Trans. Alg.*, 6(4):69, 2010. 9

[64] Chaitanya Swamy. Improved approximation algorithms for matroid and knapsack median problems and applications. *ACM Trans. Alg.*, 12(4):49, 2016. 10

[65] Paulo Rita Srgio Moro, Paulo Cortez. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 2014. 11

[66] Ke Yang and Julia Stoyanovich. Measuring fairness in ranked outputs. In *Proc. 29th International Conference on Scientific and Statistical Database Management*, page 22. ACM, 2017. 1, 3

[67] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *Proc. 20th Proceedings, International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 962–970, 2017. 1, 3

[68] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *Proceedings, International Conference on Machine Learning (ICML)*, pages 325–333, 2013. 1, 3