Fair k-Center Clustering for Data Summarization

Matthäus Kleindessner [1] **Pranjal Awasthi** [1] **Jamie Morgenstern** [2]

Abstract

In data summarization we want to choose k prototypes in order to summarize a data set. We study a setting where the data set comprises several demographic groups and we are restricted to choose k_i prototypes belonging to group i. A common approach to the problem without the fairness constraint is to optimize a centroid-based clustering objective such as k-center. A natural extension then is to incorporate the fairness constraint into the clustering problem. Existing algorithms for doing so run in time super-quadratic in the size of the data set, which is in contrast to the standard k-center problem being approximable in linear time. In this paper, we resolve this gap by providing a simple approximation algorithm for the k-center problem under the fairness constraint with running time linear in the size of the data set and k. If the number of demographic groups is small, the approximation guarantee of our algorithm only incurs a constant-factor overhead.

1. Introduction

Machine learning (ML) algorithms have been rapidly adopted in numerous human-centric domains, from personalized advertising to lending to health care. Fast on the heels of this ubiquity have come a whole host of concerning behaviors from these algorithms: facial recognition has higher accuracy on white, male faces (Buolamwini & Gebru, 2017); online advertisements suggesting arrest are shown more frequently to search queries that comprise a name primarily associated with minority groups (Sweeney, 2013); and criminal recidivism tools are likely to mislabel black low-risk defendants as high-risk while mislabeling white high-risk defendants as low-risk (Angwin et al., 2016). There are also several examples of unsavory ML behavior pertaining to unsupervised learning tasks, such as gender stereotypes in word2vec embeddings (Bolukbasi et al., 2016). Most of the academic work on fairness in ML, however, has investigated how to solve classification tasks subject to various constraints on the behavior of a classifier on different demographic groups (e.g., Hardt et al., 2016; Zafar et al., 2017).

This paper adds to the literature on fair methods for unsupervised learning tasks (see Section 4 for related work). We consider the problem of data summarization (Hesabi et al., 2015) through the lens of algorithmic fairness. The goal of data summarization is to output a small but representative subset of a data set. Think of an image database and a user entering a query that is matched by many images. Rather than presenting the user with all matching images, we only want to show a summary. In such an example, a data summary can be quite unfair on a demographic group. Indeed, Google Images has been found to answer the query "CEO" with a much higher fraction of images of men compared to the real-world fraction of male CEOs (Kay et al., 2015).

One approach to the problem of data summarization is provided by centroid-based clustering, such as k-center (formally defined in Section 2) or k-medoid (Hastie et al., 2009, Section 14.3.10; sometimes referred to as k-median). For a centroid-based clustering objective, an optimal clustering of a data set S can be defined by k points $c_1^*, \dots, c_k^* \in S$, called centroids, such that the clusters are formed by assigning every $s \in S$ to its closest centroid. Since the centroids are good representatives of their clusters, the set of centroids can be used as a summary of S. This approach of data summarization via centroid-based clustering is used in numerous domains, for example in text summarization (Moens et al., 1999) or robotics (Girdhar & Dudek, 2012).

If the data set S comprises several demographic groups $S_1, \dots, S_m$, we may consider $c_1^*, \dots, c_k^*$ to be a fair summary only if the groups are represented fairly: if in the real world 70% of CEOs are male and we want to output ten images for the query "CEO", then three of the ten images should show women. Formally, this can be encoded with one parameter k_{S_i} for every group S_i. Our goal is then to minimize the clustering objective under the constraint that k_{S_i} many centroids belong to S_i. A constraint of this form can also enforce balanced summaries: even if in the real

[1]Department of Computer Science, Rutgers University, NJ [2]College of Computing, Georgia Tech, GA. Correspondence to: Matthäus Kleindessner <matthaeus.kleindessner@rutgers.edu>, Pranjal Awasthi <pranjal.awasthi@rutgers.edu>, Jamie Morgenstern <jamiemmt@cs.gatech.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

world there are more male CEOs than female ones, we might want to output an equal number of male and female images to reflect that gender is not definitional to the role of CEO.

Centroid-based clustering under such a constraint has been studied in the theoretical computer science literature (see Sections 2 and 4). However, existing approximation algorithms for this problem run in time $\omega(|S|^2)$, while the unconstrained k-center clustering problem can be approximated in time linear in $|S|$. Since data summarization is particularly useful for massive data sets, such a slowdown may be practically prohibitive. The contribution of this paper is to present a simple approximation algorithm for k-center clustering under our fairness constraint with running time only linear in $|S|$ and k. The improved running time comes at the price of a worse guarantee on the approximation factor if the number of demographic groups is large. However, note that in practical situations concerning fairness, the number of groups is often quite small (e.g., when the groups encode gender or race). Furthermore, in our extensive numerical simulations we *never* observed a large approximation factor, even when the number of groups was large (cf. Section 5), indicating the practical usefulness of our algorithm.

Outline of the paper In Section 2, we formally state the k-center and the fair k-center problem. In Section 3, we present our algorithm and provide a sketch of its analysis. The full proofs can be found in Appendix A. We discuss related work in Section 4 and present a number of experiments in Section 5. Further experiments can be found in Appendix B. We conclude with a discussion in Section 6.

Notation For $l \in \mathbb{N}$, we sometimes use $[l] = \{1, \dots, l\}$.

2. Definition of k-Center and Fair k-Center

Let S be a finite data set and $d : S \times S \to \mathbb{R}_{\geq 0}$ be a metric on S. In particular, we assume d to satisfy the triangle inequality. The standard k-center clustering problem is the minimization problem

$$\underset{C=\{c_1,\dots,c_k\}\subseteq S}{\text{minimize}} \ \max_{s\in S} \ d(s, C), \tag{1}$$

where $k \in \mathbb{N}$ is a given parameter and $d(s, C) = \min_{c\in C} d(s, c)$. Here, $c_1, \dots, c_k$ are called centers. Any set of centers defines a clustering of S by assigning every $s \in S$ to its closest center. The k-center problem is NP-hard and is also NP-hard to approximate to a factor better than 2 (Gonzalez, 1985; Vazirani, 2001, Chapter 5). The famous greedy strategy of Gonzalez (1985) is a 2-approximation algorithm with running time $\mathcal{O}(k|S|)$ if we assume that d can be evaluated in constant time (this is the case, e.g., if a problem instance is given via the distance matrix $(d(s, s'))_{s,s'\in S}$). This greedy strategy chooses an arbitrary element of the data set as first center and then iteratively selects the data point with maximum distance to the current set of centers as the next center to be added.

Algorithm 1 Approximation algorithm for (3)

Input: metric $d : S \times S \to \mathbb{R}_{\geq 0}$; $k \in \mathbb{N}_0$; $C'_0 \subseteq S$
Output: $C = \{c_1, \dots, c_k\} \subseteq S$
set $C = \emptyset$
for $i = 1$ **to** $i = k$
  choose $c_i \in \operatorname{argmax}_{s\in S} d(s, C \cup C'_0)$
  set $C = C \cup \{c_i\}$
return C

We consider a fair variant of the k-center problem as described in Section 1. Our variant also allows for the user to specify a subset $C_0 \subseteq S$ that *has to be included* in the set of centers (think of the example of the image database and the case that we always want to show five prespecified images as part of the summary). Assuming that $S = \dot\bigcup_{i=1}^m S_i$, where $S_1, \dots S_m$ are the m demographic groups, the fair k-center problem can be stated as the minimization problem

$$\underset{\substack{C=\{c_1,\dots,c_k\}\subseteq S:\\ |C\cap S_i|=k_{S_i},\ i=1,\dots,m}}{\text{minimize}} \ \max_{s\in S} \ d(s, C \cup C_0), \tag{2}$$

where $k_{S_i} \in \mathbb{N}_0$ with $\sum_{i=1}^m k_{S_i} = k$ and $C_0 \subseteq S$ are given. By means of a partition matroid, the fair k-center problem can be phrased as a matroid center problem, for which Chen et al. (2016) provide a 3-approximation algorithm using matroid intersection (e.g., Cook et al., 1998). Chen et al. (2016) do not discuss the running time of their algorithm, but it requires to sort all distances between elements in S and hence has running time at least $\Omega(|S|^2 \log |S|)$. In our experiments in Section 5 we observe a running time in $\Omega(|S|^{5/2})$.

3. A Linear-time Approximation Algorithm

In this section, we present our approximation algorithm for the minimization problem (2). It is a recursive algorithm with respect to the number of groups m. To increase comprehensibility, we first present the case of two groups and then the general case of an arbitrary number of groups.

At several points, we will consider the standard (unfair) k-center problem (1) generalized to the case of initially given centers $C'_0 \subseteq S$, that is

$$\underset{C=\{c_1,\dots,c_k\}\subseteq S}{\text{minimize}} \ \max_{s\in S} \ d(s, C \cup C'_0). \tag{3}$$

We can adapt the greedy strategy of Gonzalez (1985) for (1) to problem (3) while maintaining its 2-approximation guarantee. For the sake of completeness, we provide the algorithm as Algorithm 1 and state the following lemma:

Lemma 1. *Algorithm 1 is a 2-approximation algorithm for the unfair k-center problem (3) with running time $\mathcal{O}((k + |C'_0|)|S|)$, assuming d can be evaluated in constant time.*

Algorithm 2 Approximation algorithm for (2) when $m = 2$

Input: metric $d : S \times S \to \mathbb{R}_{\geq 0}$; $k_{S_1}, k_{S_2} \in \mathbb{N}_0$ with $k_{S_1} + k_{S_2} = k$; $C_0 \subseteq S$; group-membership vector $\in \{1, 2\}^{|S|}$ encoding membership in S_1 or S_2
Output: $C^A = \{c_1^A, \ldots, c_k^A\} \subseteq S$
run Algorithm 1 on S with $k = k_{S_1} + k_{S_2}$ and $C_0' = C_0$; let $\widetilde{C}^A = \{\tilde{c}_1^A, \ldots, \tilde{c}_k^A\}$ denote its output

if $|\widetilde{C}^A \cap S_1| = k_{S_1}$ *# implies* $|\widetilde{C}^A \cap S_2| = k_{S_2}$
  return $\widetilde{C}^A$
 # we assume $|\widetilde{C}^A \cap S_1| > k_{S_1}$; otherwise we switch the role of S_1 and S_2
form clusters $L_1, \ldots, L_k, L_1', \ldots, L_{|C_0|}'$ by assigning every $s \in S$ to its closest center in $\widetilde{C}^A \cup C_0$
while $|\widetilde{C}^A \cap S_1| > k_{S_1}$ **and** there exists L_i with center $\tilde{c}_i^A \in S_1$ and $y \in L_i \cap S_2$
  replace center $\tilde{c}_i^A$ with y by setting $\tilde{c}_i^A = y$

if $|\widetilde{C}^A \cap S_1| = k_{S_1}$ *# implies* $|\widetilde{C}^A \cap S_2| = k_{S_2}$
  return $\widetilde{C}^A$
let $S' = \cup_{i \in [k]: \tilde{c}_i^A \in S_1} L_i$ *# we have $S' \subseteq S_1$*
run Algorithm 1 on $S' \cup C_0'$ with $k = k_{S_1}$ and $C_0' = C_0 \cup (\widetilde{C}^A \cap S_2)$; let $\widehat{C}^A$ denote its output
return $\widehat{C}^A \cup (\widetilde{C}^A \cap S_2)$ as well as $(k_{S_2} - |\widetilde{C}^A \cap S_2|)$ many arbitrary elements from S_2

A proof of Lemma 1, similar in structure to a proof in Har-Peled (2011, Section 4.2) for the strategy of Gonzalez (1985) for problem (1), can be found in Appendix A.

3.1. Fair k-Center with Two Groups

Assume that $S = S_1 \dot{\cup} S_2$. Our algorithm first runs Algorithm 1 for the unfair problem (3) with $k = k_{S_1} + k_{S_2}$ and $C_0' = C_0$. If we are lucky and Algorithm 1 picks k_{S_1} many centers from S_1 and k_{S_2} many centers from S_2, our algorithm terminates. Otherwise, Algorithm 1 picks too many centers from one group, say S_1, and too few from S_2. We try to decrease the number of centers in S_1 by replacing any such a center with an element in its cluster belonging to S_2. Once we have made all such available swaps, the remaining clusters with centers in S_1 are entirely contained within S_1. We then run Algorithm 1 on these clusters with $k = k_{S_1}$ and the centers from S_2 as well as C_0 as initially given centers, and return both the centers from the recursive call (all in S_1) and those from the initial call and the swapping in S_2.

This algorithm is formally stated as Algorithm 2. The following theorem states that it is a 5-approximation algorithm and that our analysis is tight—in general, Algorithm 2 does not achieve a better approximation factor.

Theorem 1. *Algorithm 2 is a 5-approximation algorithm for the fair k-center problem (2) with $m = 2$, but not a $(5 - \varepsilon)$-approximation algorithm for any $\varepsilon > 0$. It can be implemented in time $\mathcal{O}((k + |C_0|)|S|)$, assuming d can be evaluated in constant time.*

Proof. Here we only present a sketch of the proof. The full proof can be found in Appendix A. For showing that Algorithm 2 is a 5-approximation algorithm, let r^*_{fair} be the optimal value of (2) and r^* be the optimal value of (3) (for $C_0' = C_0$). Clearly, $r^* \leq r^*_{\text{fair}}$. Let C^A be the set of centers returned by Algorithm 2. It is clear that C^A comprises k_{S_1} many elements from S_1 and k_{S_2} many elements from S_2. We need to show that $\min_{c \in C^A \cup C_0} d(s, c) \leq 5r^*_{\text{fair}}$ for every $s \in S$. Let $\widetilde{C}^A$ be the output of Algorithm 1 when called in Line 3 of Algorithm 2. Since Algorithm 1 is a 2-approximation algorithm for (3) according to Lemma 1, we have $\min_{c \in \widetilde{C}^A \cup C_0} d(s, c) \leq 2r^* \leq 2r^*_{\text{fair}}$, $s \in S$. Assume that $|\widetilde{C}^A \cap S_1| > k_{S_1}$. It follows from the triangle inequality that after exchanging centers in the while-loop in Line 9 of Algorithm 2 we have $\min_{c \in \widetilde{C}^A \cup C_0} d(s, c) \leq 4r^*_{\text{fair}}$, $s \in S$. Assume that still $|\widetilde{C}^A \cap S_1| > k_{S_1}$. We only need to show that $\min_{c \in C^A \cup C_0} d(s, c) \leq 5r^*_{\text{fair}}$ for $s \in S'$. Let C^*_{fair} be an optimal solution to (2). We split S' into two subsets $S' = S'_a \dot{\cup} S'_b$, where S'_a comprises all $s \in S'$ for which the closest center in $C^*_{\text{fair}} \cup C_0$ is in $S_2 \cup C_0$. Using the triangle inequality we can show that $\min_{c \in C^A \cup C_0} d(s, c) \leq 5r^*_{\text{fair}}$, $s \in S'_a$. We partition S'_b into at most k_{S_1} many clusters corresponding to the closest center in C^*_{fair}. Each of these clusters has diameter not greater than $2r^*_{\text{fair}}$. If Algorithm 1 in Line 15 of Algorithm 2 chooses one element from each of these clusters, we immediately have $\min_{c \in C^A \cup C_0} d(s, c) \leq 2r^*_{\text{fair}}$, $s \in S'_b$. Otherwise, Algorithm 1 chooses an element from S'_a or two elements from the same cluster of S'_b. In both cases, it follows from the greedy choice property of Algorithm 1 that $\min_{c \in C^A \cup C_0} d(s, c) \leq 5r^*_{\text{fair}}$, $s \in S'_b$.

A family of examples shows that Algorithm 2 is not a $(5-\varepsilon)$-approximation algorithm for any $\varepsilon > 0$. □

3.2. Fair k-Center with Arbitrary Number of Groups

The main idea to handle an arbitrary number of groups m is the same as for the case $m = 2$: we first run Algorithm 1. We then exchange centers for elements in their clusters in such a way that the number of centers from a group S_i comes closer to k_{S_i}, which is the requested number of centers from S_i. If via exchanging centers we can actually hit k_{S_i} for every group S_i, we are done. Otherwise, we wish that, when no more exchanging is possible, we are left with a subset $S' \subseteq S$ that only comprises elements from $m - 1$ or fewer groups. Denote the set of these groups by $\mathcal{G}$. We also wish that for those groups not in $\mathcal{G}$ we have picked only the requested number of centers or fewer and we can consider the groups not in $\mathcal{G}$ to have been "resolved". If both are true,

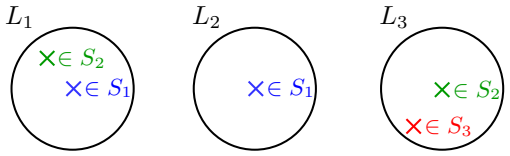

Figure 1. An example illustrating the need for a more sophisticated procedure for exchanging centers in the case of three or more groups compared to the case of only two groups: we would like to exchange a center from S_1 for an element from S_3, but cannot do that directly. Rather, we have to make a series of exchanges.

we can recursively apply our algorithm to S' and a smaller number of groups. We might recurse down to the case of only one group, which we can solve with Algorithm 1.

The difficulty with this idea comes from the exchanging process. Formally, we are given k centers $\tilde{c}_1^A, \dots, \tilde{c}_k^A$ and the corresponding clustering $S \setminus S_{C_0} = \dot{\bigcup}_{i=1}^{k} L_i$, where $S_{C_0} = \dot{\bigcup}_{i=1}^{|C_0|} L_i'$ is the union of clusters with a center in C_0, and we want to exchange some centers $\tilde{c}_i^A$ for an element in their cluster L_i such that there exists a strict subset of groups $\mathcal{G} \subsetneq \{S_1, \dots, S_m\}$ with the following properties:

$$\bigcup_{i \in [k]:\, \tilde{c}_i^A \text{ is from a group in } \mathcal{G}} L_i \subseteq \bigcup_{S_i \in \mathcal{G}} S_i, \quad (4)$$

$$\forall S_j \in \{S_1, \dots, S_m\} \setminus \mathcal{G} : \sum_{i=1}^{k} \mathbb{1}\left\{\tilde{c}_i^A \in S_j\right\} \leq k_{S_j}. \quad (5)$$

While in the case of only two groups this can easily be achieved by exchanging centers from the group that has more than the requested number of centers for elements from the other group, as we do in Algorithm 2, it is not immediately clear how to deal with a situation as shown in Figure 1. There are three groups S_1, S_2, S_3 (elements of these groups are shown in blue, green, and red, respectively), and we have $k_{S_1} = k_{S_2} = k_{S_3} = 1$. For the current set of centers (elements at the centers of the circles) there does not exist $\mathcal{G} \subsetneq \{S_1, S_2, S_3\}$ satisfying (4) and (5). We would like to decrease the number of centers in S_1 and increase the number of centers in S_3, but the clusters with a center in S_1 do not comprise an element from S_3. Hence, we cannot directly exchange a center from S_1 for an element in S_3. Rather, we first have to exchange a center from S_1 for an element in S_2 (although this increases the number of centers from S_2 over k_{S_2}) and then a center from S_2 for an element in S_3. An algorithm that can deal with such a situation is Algorithm 3. It exchanges some centers for an element in their cluster L_i and yields $\mathcal{G} \subsetneq \{S_1, \dots, S_m\}$ that provably satisfies (4) and (5), as stated by the following lemma. Its proof can be found in Appendix A.

Lemma 2. *Algorithm 3 is well-defined, it terminates, and exchanges centers in such a way that the set $\mathcal{G}$ that it returns satisfies $\mathcal{G} \subsetneq \{S_1, \dots, S_m\}$ and properties (4) and (5).*

Algorithm 3 Algorithm for exchanging centers & finding $\mathcal{G}$

Input: centers $\tilde{c}_1^A, \dots, \tilde{c}_k^A$ and the corresponding clustering $S \setminus S_{C_0} = \dot{\bigcup}_{i=1}^{k} L_i$; $k_{S_1}, \dots, k_{S_m} \in \mathbb{N}_0$ with $\sum_{i=1}^{m} k_{S_i} = k$; group-membership vector $\in \{1, \dots, m\}^{|S \setminus S_{C_0}|}$
Output: $\tilde{c}_1^A, \dots, \tilde{c}_k^A$, where some centers $\tilde{c}_i^A$ have been replaced with an element in L_i, and $\mathcal{G} \subsetneq \{S_1, \dots, S_m\}$ satisfying (4) and (5)

set $\tilde{k}_{S_j} = \sum_{i=1}^{k} \mathbb{1}\left\{\tilde{c}_i^A \in S_j\right\}$ for $S_j \in \{S_1, \dots, S_m\}$
construct a directed unweighted graph G on $V = \{S_1, \dots, S_m\}$ as follows: we have $S_i \to S_j$, that is there is a directed edge from S_i to S_j, if and only if there exists L_t with center $\tilde{c}_t^A \in S_i$ and $y \in L_t \cap S_j$
compute all shortest paths on G

while $\tilde{k}_{S_j} \neq k_{S_j}$ for some S_j and there exist S_r, S_s such that $\tilde{k}_{S_r} > k_{S_r}$ and $\tilde{k}_{S_s} < k_{S_s}$ and there exists a shortest path $P = S_{v_0} S_{v_1} \cdots S_{v_w}$ with $S_{v_0} = S_r$, $S_{v_w} = S_s$ that connects S_r to S_s in G
  for $l = 0, \dots, w-1$
    find L_t with center $\tilde{c}_t^A \in S_{v_l}$ and $y \in L_t \cap S_{v_{l+1}}$; replace $\tilde{c}_t^A$ with y by setting $\tilde{c}_t^A = y$
  update $\tilde{k}_{S_r} = \tilde{k}_{S_r} - 1$ and $\tilde{k}_{S_s} = \tilde{k}_{S_s} + 1$
  recompute G and all shortest paths on G

if $\tilde{k}_{S_j} = k_{S_j}$ for all $S_j \in \{S_1, \dots, S_m\}$
  return $\tilde{c}_1^A, \dots, \tilde{c}_k^A$ and $\mathcal{G} = \emptyset$
else
  set $\mathcal{G}' = \{S_j \in \{S_1, \dots, S_m\} : \tilde{k}_{S_j} > k_{S_j}\}$ and $\mathcal{G} = \mathcal{G}' \cup \{S_j \in \{S_1, \dots, S_m\} \setminus \mathcal{G}' :$ there exists $S_i \in \mathcal{G}'$ and a path from S_i to S_j in $G\}$
  return $\tilde{c}_1^A, \dots, \tilde{c}_k^A$ and $\mathcal{G}$

Observing that the number of iterations of the while-loop in Line 7 is upper-bounded by k as the proof of Lemma 2 shows, that the number of iterations of the for-loop in Line 8 is upper-bounded by m, and that all shortest paths on G can be computed in running time $\mathcal{O}(m^3)$ (Cormen et al., 2009, Chapter 25), it is not hard to see that Algorithm 3 can be implemented with running time $\mathcal{O}(km|S| + km^3)$.

Using Algorithm 3, it is straightforward to design a recursive approximation algorithm for the fair k-center problem (2) as outlined at the beginning of Section 3.2. We state the algorithm as Algorithm 4. Applying, by means of induction, a similar technique as in the proof of Theorem 1 to every (recursive) call of Algorithm 4, we can prove the following:

Theorem 2. *Algorithm 4 is a $(3 \cdot 2^{m-1} - 1)$-approximation algorithm for the fair k-center problem (2) with m groups. It can be implemented in time $\mathcal{O}((|C_0|m + km^2)|S| + km^4)$, assuming d can be evaluated in constant time.*

Algorithm 4 Approximation alg. for (2) for arbitrary m

Input: metric $d : S \times S \to \mathbb{R}_{\geq 0}$; $k_{S_1}, \dots, k_{S_m} \in \mathbb{N}_0$ with $\sum_{i=1}^m k_{S_i} = k$; $C_0 \subseteq S$; group-membership vector $\in \{1, \dots, m\}^{|S|}$
Output: $C^A = \{c_1^A, \dots, c_k^A\} \subseteq S$
run Algorithm 1 on S with $k = \sum_{i=1}^m k_{S_i}$ and $C_0' = C_0$; let $\widetilde{C}^A = \{\tilde{c}_1^A, \dots, \tilde{c}_k^A\}$ denote its output
if $m = 1$
  return $\widetilde{C}^A$

form clusters $L_1, \dots, L_k, L_1', \dots, L_{|C_0|}'$ by assigning every $s \in S$ to its closest center in $\widetilde{C}^A \cup C_0$
apply Algorithm 3 to $\tilde{c}_1^A, \dots, \tilde{c}_k^A$ and $\dot{\cup}_{i=1}^k L_i$ in order to exchange some centers $\tilde{c}_i^A$ and obtain $\mathcal{G} \subsetneq \{S_1, \dots, S_m\}$
if $\mathcal{G} = \emptyset$
  return $\widetilde{C}^A$

let $S' = \cup_{i \in [k]:\, \tilde{c}_i^A \text{ is from a group in } \mathcal{G}} L_i$ and $C' = \{\tilde{c}_i^A \in \widetilde{C}^A : \tilde{c}_i^A \text{ is from a group not in } \mathcal{G}\}$; recursively call Algorithm 4, where:

- $S' \cup C' \cup C_0$ plays the role of S
- we assign elements in $C' \cup C_0$ to an arbitrary group in $\mathcal{G}$ and hence there are $|\mathcal{G}| < m$ many groups $S_{j_1}, \dots, S_{j_{|\mathcal{G}|}}$
- the requested numbers of centers are $k_{S_{j_1}}, \dots, k_{S_{j_{|\mathcal{G}|}}}$
- $C' \cup C_0$ plays the role of initially given centers C_0

let $\widehat{C}^R$ denote its output
return $\widehat{C}^R \cup C'$ as well as $(k_{S_j} - |C' \cap S_j|)$ many arbitrary elements from S_j for every group S_j not in $\mathcal{G}$

It is not clear to us whether our analysis of Algorithm 4 is tight and the approximation factor achieved by Algorithm 4 can indeed be as large as $(3 \cdot 2^{m-1} - 1)$ or whether the dependence on m is actually less severe (compare with Section 5 and Section 6). Although trying hard to find instances for which the approximation factor of Algorithm 4 is large, we never observed a factor greater than 8.

Lemma 3. *Algorithm 4 is not a $(8 - \varepsilon)$-approximation algorithm for any $\varepsilon > 0$ for (2) with $m \geq 3$ groups.*

The proofs of Theorem 2 and Lemma 3 are in Appendix A.

4. Related Work

Fairness By now, there is a huge body of work on fairness in machine learning. For a recent paper providing an overview of the literature on fair classification see Donini et al. (2018). Our paper adds to the literature on fair methods for unsupervised learning tasks (Chierichetti et al., 2017; Celis et al., 2018a;b;c; Samadi et al., 2018; Schmidt et al., 2018). Note that all these papers assume to know which demographic group a data point belongs to just as we do. We discuss the two works most closely related to our paper.

First, Celis et al. (2018b) also deal with the problem of fair data summarization. They study the same fairness constraint as we do, that is the summary must contain k_{S_i} many elements from group S_i. However, while we aim for a *representative* summary, where every data point should be close to at least one center in the summary, Celis et al. aim for a *diverse* summary. Their approach requires the data set S to consist of points in $\mathbb{R}^n$, and then the diversity of a subset of S is measured by the volume of the parallelepiped that it spans (Kulesza & Taskar, 2012). This summarization objective is different from ours, and in different applications one or the other may be more appropriate. An advantage of our approach is that it only requires access to a metric on the data set rather than feature representations of data points.

The second line of work we discuss centers around the paper of Chierichetti et al. (2017). Their paper proposes a notion of fairness for clustering different from ours. Based on the fairness notion of disparate impact (Feldman et al., 2015) / the $p\%$-rule (Zafar et al., 2017) for classification, the paper by Chierichetti et al. asks that every group be approximately equally represented in each cluster. In their paper, Chierichetti et al. focus on k-medoid and k-center clustering and the case of two groups. Subsequently, Rösner & Schmidt (2018) study such a fair k-center problem for multiple groups, and Schmidt et al. (2018) build upon the work of Chierichetti et al. to devise algorithms for such a fair k-means problem. Kleindessner et al. (2019) incorporate the fairness notion of Chierichetti et al. into the spectral clustering framework. While we certainly consider the fairness notion of Chierichetti et al. (2017), which can be applied to any kind of clustering, to be meaningful in some scenarios, we believe that in certain applications of centroid-based clustering (such as data summarization) our proposed fairness notion provides a more sensible alternative.

Centroid-based clustering There are many papers proposing heuristics and approximation algorithms for both k-center (e.g., Hochbaum & Shmoys, 1986; Mladenović et al., 2003; Ferone et al., 2017) and k-medoid (e.g., Charikar et al., 2002; Arya et al., 2004; Li & Svensson, 2013) under various assumptions on S and the distance function d. There are also numerous papers on versions with constraints, such as lower or upper bounds on the size of the clusters (Aggarwal et al., 2010; Cygan et al., 2012; Rösner & Schmidt, 2018).

Most important to mention are the works by Hajiaghayi et al. (2010), Krishnaswamy et al. (2011) and Chen et al. (2016). Hajiaghayi et al. are the first that consider our fairness constraint (for two groups) for k-medoid. They present

a local search algorithm and prove it to be a constant-factor approximation algorithm. Their work has been generalized by Krishnaswamy et al., who consider k-medoid under the constraint that the centers have to form an independent set in a given matroid. This kind of constraint contains our fairness constraint as a special case (for an arbitrary number of groups). Krishnaswamy et al. obtain a 16-approximation algorithm for this so-called matroid median problem based on rounding the solution of a linear programming relaxation. Subsequently, Chen et al. study the matroid center problem. Using matroid intersection as black box, they obtain a 3-approximation algorithm. Note that none of Hajiaghayi et al., Krishnaswamy et al. or Chen et al. discuss the running time of their algorithm, except for arguing it to be polynomial (see Section 2). We also mention the works by Chakrabarty & Negahbani (2018), who provide a generalization of the matroid center problem and in doing so recover the result of Chen et al. (2016), and by Kale (2018), who studies the matroid center problem in a streaming setting.

5. Experiments

In this section, we present a number of experiments[1]. We begin with a motivating example on a small image data set illustrating that a summary produced by Algorithm 1 (i.e., the standard greedy strategy for the unfair k-center problem) can be quite unfair. We also compare summaries produced by our algorithm to summaries produced by the method of Celis et al. (2018b). We then investigate the approximation factor of our algorithm on several artificial instances with known or computable optimal value of the fair k-center problem (2) and compare our algorithm to the one for the matroid center problem by Chen et al. (2016), both in terms of approximation factor / cost of output and running time. Next, on both synthetic and real data, we compare our algorithm in terms of the cost of its output to two baseline heuristics (with running time linear in $|S|$ and k just as for our algorithm). Finally, we compare our algorithm to Algorithm 1 more systematically. We study the difference in the costs of the outputs of our algorithm and Algorithm 1, a quantity one may refer to as *price of fairness*, and measure how unfair the output of Algorithm 1 can be. In the following, all boxplots show results of 200 runs of an experiment.

5.1. Motivating Example and Comparison with Celis et al. (2018b)

Consider the 14 images[2] of medical doctors shown in the first row of Figure 2. Assume we want to generate a summary of size four of these images. One way to do so is to run Algorithm 1. The first column of the table in Figure 2 shows in each row the summary produced in one run of Algorithm 1 (recall that all algorithms considered here are randomized algorithms). These summaries are quite unfair: although there is an equal number of images of female doctors and images of male doctors, all these summaries show three or even four females. To overcome this bias we can apply our algorithm or the method of Celis et al. (2018b), which both allow us to explicitly state the numbers of females and males that we want in the summary. The second and the third column of the table show summaries produced by these algorithms. It is hard to say which of them produces more useful summaries and the results ultimately depend on the feature representations of the images (see the next paragraph). To provide further illustration, we present a similar experiment in Figure 11 in Appendix B.

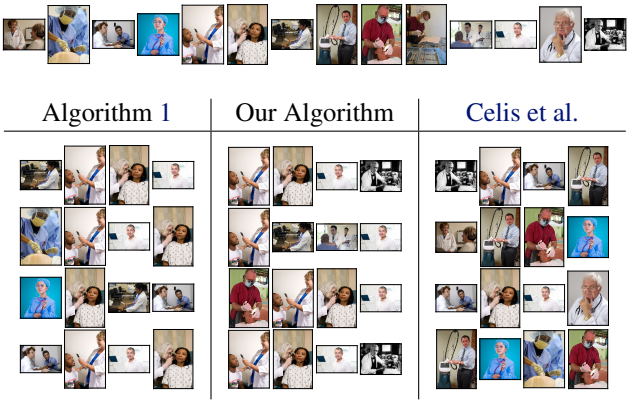

Figure 2. A data set consisting of 14 images of medical doctors (7 female, 7 male) and four summaries computed by the unfair Algorithm 1, our algorithm and the algorithm proposed by Celis et al. (2018b) (all three algorithms are randomized algorithms).

For computing feature representations of the images and running the algorithm of Celis et al. we used the code provided by them. The feature vector of an image is a histogram based on the image's SIFT descriptors; see Celis et al. for details. We used the Euclidean metric between these feature vectors as metric d for Algorithm 1 and our algorithm.

5.2. Approximation Factor and Comparison with Chen et al. (2016)

We implemented the algorithm by Chen et al. (2016) using the generic algorithm for matroid intersection provided in SageMath[3]. To speed up computation, rather than testing all distance values as threshold as suggested by Chen et al., we implemented binary search to look for the optimal value.

In the experiment shown in the left part of Figure 3, we study

[1] Python code is available on https://github.com/matthklein/fair_k_center_clustering.

[2] All images were found on https://pexels.com, https://pixnio.com or https://commons.wikimedia.org and are in the public domain.

[3] http://sagemath.org/

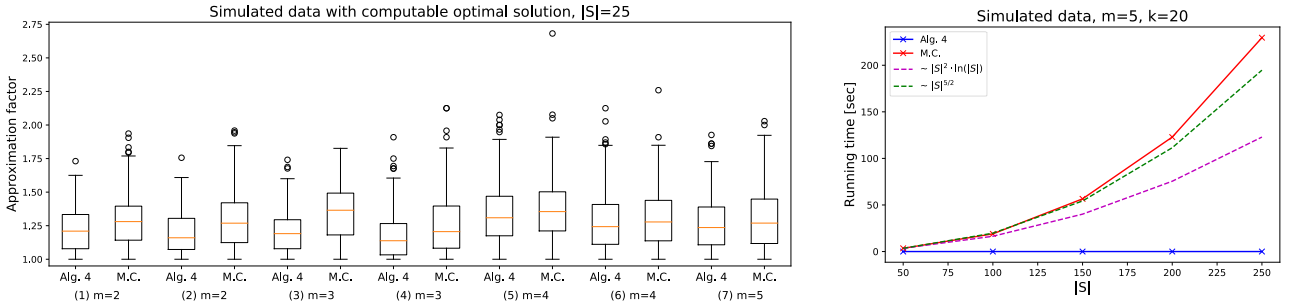

Figure 3. **Left:** Approx. factor of Alg. 4 and the algorithm by Chen et al. (M.C.) on simulated data with computable optimal solution. $|S| = 25$; various settings with $m \in \{2, 3, 4, 5\}$. **(1)** $|C_0| = 2$, $(k_{S_1}, k_{S_2}) = (2, 2)$ **(2)** $|C_0| = 2$, $(k_{S_1}, k_{S_2}) = (4, 2)$ **(3)** $|C_0| = 2$, $(k_{S_1}, k_{S_2}, k_{S_3}) = (2, 2, 2)$ **(4)** $|C_0| = 1$, $(k_{S_1}, k_{S_2}, k_{S_3}) = (5, 1, 1)$ **(5)** $C_0 = \emptyset$, $(k_{S_1}, k_{S_2}, k_{S_3}, k_{S_4}) = (2, 2, 2, 2)$ **(6)** $C_0 = \emptyset$, $(k_{S_1}, k_{S_2}, k_{S_3}, k_{S_4}) = (3, 3, 1, 1)$ **(7)** $C_0 = \emptyset$, $(k_{S_1}, k_{S_2}, k_{S_3}, k_{S_4}, k_{S_5}) = (2, 2, 2, 1, 1)$. **Right:** Running time as a function of $|S|$.

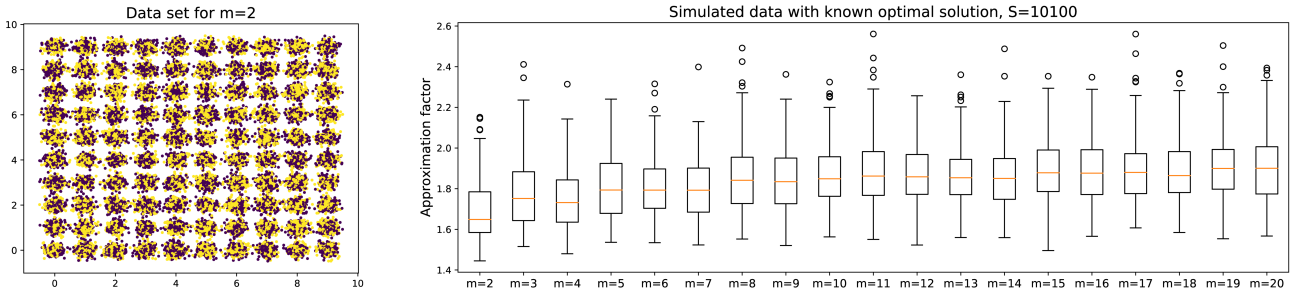

Figure 4. Approximation factor of our algorithm on simulated data with known optimal solution. $|S| = 10100$, $C_0 = \emptyset$, $\sum_{i=1}^{m} k_{S_i} = 100$. **Left:** Example of the data set when $m = 2$. The optimal solution consists of 100 points located at the centers of the visible clusters and has cost 0.5. **Right:** Approximation factor for $m \in \{2, \ldots, 20\}$.

the approximation factor achieved by our algorithm (Alg. 4) and the algorithm by Chen et al. (M.C.) in various settings of values of m, $|C_0|$ and k_{S_i}, $i \in [m]$. The data set S always consists of 25 vertices of a random graph and is small enough to explicitly compute an optimal solution to the fair k-center problem (2). The random graph is constructed according to an Erdős-Rényi model, where any possible edge between two vertices is contained in the graph with probability $2\log(|S|)/|S|$. With high probability such a graph is connected (if not, we discard it). We put random weights on the edges, drawn from the uniform distribution on $[100]$, and let the metric d be the shortest-path distance on the graph. We assign every vertex to one of m groups uniformly at random and randomly choose a subset $C_0 \subseteq S$ of initially given centers. As we can see from the boxplots, the approximation factor achieved by our algorithm is *never* larger than 2.2. We also see that in each of the seven settings that we consider the median of the achieved approximation factors (indicated by the red lines in the boxes) is smaller for our algorithm than for the algorithm by Chen et al..

In the experiment shown in the right part of Figure 3, we study the running time of the two algorithms as a function of the size of the data set, which is created analogously to the experiment in the left part. We set $m = 5$, $C_0 = \emptyset$ and $k_{S_i} = 4$, $i \in [5]$. The shown curves are obtained from averaging the running times of 200 runs of the experiment (performed on an iMac with 3.4 GHz i5 / 8 GB DDR4). While our algorithm never runs for more than 0.01 seconds, the algorithm by Chen et al., on average, runs for 230 seconds when $|S| = 250$. Its run time grows at least as $|S|^{5/2}$, which proves it to be inappropriate for massive data sets. Boxplots of the costs of the outputs obtained in this experiment are provided in Figure 9 in Appendix B. We can see there that the costs are very similar for the two algorithms.

In the experiment of Figure 4, we once more study the approximation factor achieved by our algorithm. We place 100 optimal centers at $(i, j) \in \mathbb{R}^2$, $i, j \in \{0, \ldots, 9\}$, and sample 10000 points around them such that for every center the farthest point in its cluster is at distance 0.5 from the center (Euclidean distance). One such a point set can be seen in the left plot of Figure 4. We randomly assign every point and center to one of m groups and set k_{S_i} to the number of centers that have been assigned to group S_i. We let $C_0 = \emptyset$. For $m \in \{2, \ldots, 20\}$, the right part of Figure 4 shows boxplots of the approximation factors for our algorithm. Similarly as before, the approximation factor achieved by our algorithm is *never* larger than 2.6. Most interestingly, the approximation factor increases very moderately with m.

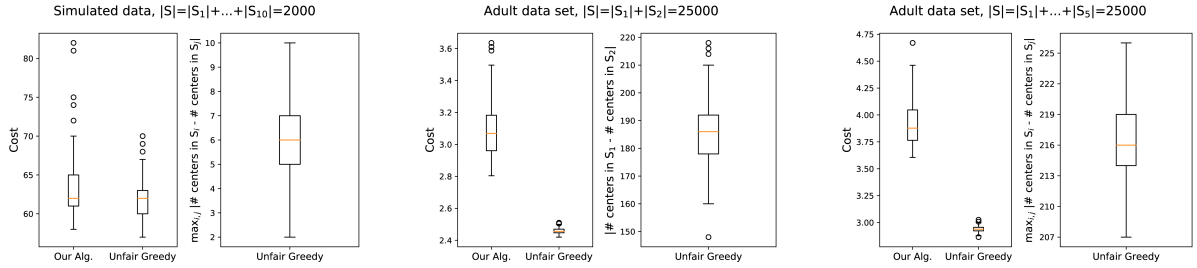

Figure 5. Cost of the output of our algorithm in comparison to the unfair Algorithm 1 and maximum deviation of the numbers of centers in S_i and S_j, $i, j \in [m]$, in the output of Algorithm 1 (it is $k_{S_i} = k_{S_j}, i, j \in [m]$). **Left:** $m = 10$. **Middle:** $m = 2$. **Right:** $m = 5$.

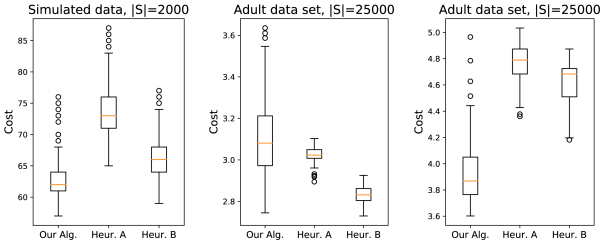

Figure 6. Cost of the output of our algorithm in comparison to two heuristics. **Left:** $m = 10$. **Middle:** $m = 2$. **Right:** $m = 5$.

5.3. Comparison with Baseline Approaches

We compare our algorithm in terms of the cost of an approximate solution to two linear-time baseline heuristics for the fair k-center problem (2). The first one, referred to as Heuristic A, runs Algorithm 1 on each group separately (with $k = k_{S_i}$ and $C_0' = S_i \cap C_0$ for group S_i) and outputs the union of the centers obtained for the groups. The second one, Heuristic B, greedily chooses centers similarly to Algorithm 1, but only from those groups for which we have not reached the requested number of centers yet. It is easy to see that the approximation factor achieved by these heuristics can be arbitrarily large on some worst-case instances.

Figure 6 shows boxplots of the costs of the approximate solutions returned by our algorithm and the two heuristics for three data sets: the data set in the left plot consists of 2000 vertices of a random graph constructed similarly as in the experiments of Figure 3. We set $m = 10$, $k_{S_i} = 4$, $i \in [10]$, and $|C_0| = 10$. The data set in the middle and in the right plot consists of the first 25000 records of the Adult data set (Dua & Graff, 2019). We only use its six numerical features (e.g., age, hours worked per week), normalized to zero mean and unit variance, for representing records and use the l_1-distance as metric d. For the experiment shown in the middle plot, we split the data set into two groups according to the sensitive feature gender (#Female=8291, #Male=16709) and set $k_{S_1} = k_{S_2} = 200$. For the experiment shown in the right plot, we split the data set into five groups according to the feature race (#White=21391, #Asian-Pac-Islander=775, #Amer-Indian-Eskimo=241, #Other=214, #Black=2379) and set $k_{S_i} = 50$, $i \in [5]$. In Figure 10 in Appendix B we present results for other choices of k_{S_i}. We always let C_0 be a randomly chosen subset of size $|C_0| = 100$. The two heuristics perform surprisingly well. Although coming without any worst-case guarantees, the cost of their solutions is comparable to the cost of the output of our algorithm.

5.4. Comparison with Unfair Algorithm 1

We compare the cost of the solution produced by our algorithm to the cost of the (potentially) unfair solution provided by Algorithm 1. Of course, we expect the latter to be lower. We consider the case $k_{S_i} = k_{S_j}, i, j \in [m]$, and also examine how balanced the numbers of centers from a group S_i in the output of Algorithm 1 are. Figure 5 shows the results, where the data sets and settings equal the ones in the experiments of Figure 6. Similar experiments with different settings are provided in Figure 12 in Appendix B. Remarkably, the costs of the solutions produced by our algorithm and Algorithm 1 have the same order of magnitude in all experiments, showing that the price of fairness is small. On the other hand, the output of Algorithm 1 can be highly unfair.

6. Discussion

In this work, we considered k-center clustering under a fairness constraint that is motivated by the application of centroid-based clustering for data summarization. We presented a simple approximation algorithm with running time only linear in the size of the data set S and the number of centers k and proved our algorithm to be a 5-approximation algorithm when S consists of two groups. For more than two groups, we proved an upper bound on the approximation factor that increases exponentially with the number of groups. We do not know whether this exponential dependence is necessary or whether our analysis is loose—in our extensive numerical simulations we *never* observed a large approximation factor. Besides answering this question, in future work it would be interesting to extend our results to k-medoid clustering or to characterize properties of data sets that guarantee that fast algorithms find an optimal fair clustering.

Acknowledgements

This research is supported by a Rutgers Research Council Grant and a Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) postdoctoral fellowship.

References

Aggarwal, G., Panigrahy, R., Feder, T., Thomas, D., Kenthapadi, K., Khuller, S., and Zhu, A. Achieving anonymity via clustering. *ACM Transactions on Algorithms*, 6(3): 49:1–49:19, 2010.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. Propublica—machine bias, 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., and Pandit, V. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004.

Bolukbasi, T., Chang, K.-W., Zou, J., Saligrama, V., and Kalai, A. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Neural Information Processing Systems (NIPS)*, 2016.

Buolamwini, J. and Gebru, T. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability, and Transparency (ACM FAT)*, 2017.

Celis, L. E., Huang, L., and Vishnoi, N. K. Multiwinner voting with fairness constraints. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018a.

Celis, L. E., Keswani, V., Straszak, D., Deshpande, A., Kathuria, T., and Vishnoi, N. K. Fair and diverse DPP-based data summarization. In *International Conference on Machine Learning (ICML)*, 2018b. Code available on https://github.com/DamianStraszak/FairDiverseDPPSampling.

Celis, L. E., Straszak, D., and Vishnoi, N. K. Ranking with fairness constraints. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2018c.

Chakrabarty, D. and Negahbani, M. Generalized center problems with outliers. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018.

Charikar, M., Guha, S., Tardos, E., and Shmoys, D. B. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65 (1):129–149, 2002.

Chen, D. Z., Li, J., Liang, H., and Wang, H. Matroid and knapsack center problems. *Algorithmica*, 75:27–52, 2016.

Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. In *Neural Information Processing Systems (NIPS)*, 2017.

Cook, W. J., Cunningham, W. H., Pulleyblank, W. R., and Schrijver, A. *Combinatorial Optimization*. Wiley, 1998.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.

Cygan, M., Hajiaghayi, M., and Khuller, S. LP rounding for k-centers with non-uniform hard capacities. In *Symposium on Foundations of Computer Science (FOCS)*, 2012.

Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J., and Pontil, M. Empirical risk minimization under fairness constraints. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Dua, D. and Graff, C. UCI machine learning repository, 2019. https://archive.ics.uci.edu/ml/datasets/adult.

Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

Ferone, D., Festa, P., Napoletano, A., and Resende, M. G. C. A new local search for the p-center problem based on the critical vertex concept. In *International Conference on Learning and Intelligent Optimization (LION)*, 2017.

Girdhar, Y. and Dudek, G. Efficient on-line data summarization using extremum summaries. In *International Conference on Robotics and Automation (ICRA)*, 2012.

Gonzalez, T. F. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38: 293–306, 1985.

Hajiaghayi, M., Khandekar, R., and Kortsarz, G. The red-blue median problem and its generalization. In *European Symposium on Algorithms (ESA)*, 2010.

Har-Peled, S. *Geometric approximation algorithms*. American Mathematical Society, 2011.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *Neural Information Processing Systems (NIPS)*, 2016.

Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning — Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

Hesabi, Z. R., Tari, Z., Goscinski, A., Fahad, A., Khalil, I., and Queiroz, C. Data summarization techniques for big data—a survey. In *Handbook on Data Centers*, pp. 1109–1152. Springer, 2015.

Hochbaum, D. S. and Shmoys, D. B. A unified approach to approximation algorithms for bottleneck problems. *Journal of the ACM*, 33(3):533–550, 1986.

Kale, S. Small space stream summary for matroid center. arXiv:1810.06267 [cs.DS], 2018.

Kay, M., Matuszek, C., and Munson, S. A. Unequal representation and gender stereotypes in image search results for occupations. In *Conference on Human Factors in Computing Systems (CHI)*, 2015.

Kleindessner, M., Samadi, S., Awasthi, P., and Morgenstern, J. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning (ICML)*, 2019.

Krishnaswamy, R., Kumar, A., Nagarajan, V., Sabharwal, Y., and Saha, B. The matroid median problem. In *Symposium on Discrete Algorithms (SODA)*, 2011.

Kulesza, A. and Taskar, B. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5:123–286, 2012.

Li, S. and Svensson, O. Approximating k-median via pseudo-approximation. In *Symposium on the Theory of Computing (STOC)*, 2013.

Mladenović, N., Labbé, M., and Hansen, P. Solving the p-center problem with tabu search and variable neighborhood search. *Networks*, 42(1):48–64, 2003.

Moens, M.-F., Uyttendaele, C., and Dumortier, J. Abstracting of legal cases: The potential of clustering based on the selection of representative objects. *Journal of the American Society for Information Science*, 50(2):151–161, 1999.

Rösner, C. and Schmidt, M. Privacy preserving clustering with constraints. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, 2018.

Samadi, S., Tantipongpipat, U., Morgenstern, J., Singh, M., and Vempala, S. The price of fair PCA: One extra dimension. In *Neural Information Processing Systems (NeurIPS)*, 2018.

Schmidt, M., Schwiegelshohn, C., and Sohler, C. Fair coresets and streaming algorithms for fair k-means clustering. arXiv:1812.10854 [cs.DS], 2018.

Sweeney, L. Discrimination in online ad delivery. *Queue*, 11(3):10–29, 2013.

Vazirani, V. *Approximation Algorithms*. Springer, 2001.

Zafar, M. B., Valera, I., Rodriguez, M. G., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.