Fair k-center Clustering with Outliers

Daichi Amagata
Osaka University

Abstract

The importance of dealing with big data is further increasing, as machine learning (ML) systems obtain useful knowledge from big datasets. However, using all data is practically prohibitive because of the massive sizes of the datasets, so summarizing them by centers obtained from k-center clustering is a promising approach. We have two concerns here. One is fairness, because if the summary does not have some specific groups, subsequent applications may provide unfair results for the groups. The other is the presence of outliers, and if outliers dominate the summary, it cannot be useful. To overcome these concerns, we address the problem of fair k-center clustering with outliers. Although prior works studied the fair k-center clustering problem, they do not consider outliers. This paper yields a linear time algorithm that satisfies the fairness constraint of our problem and probabilistically guarantees the almost 3-approximation bound. Its empirical efficiency and effectiveness are also reported.

1 Introduction

Dealing with big data is still a challenging problem, as it is inefficient or prohibitive to use all data in a given dataset, e.g., for training ML models. Data summarization is an effective solution to this concern, because we can control the size of a summary based on the requirements of applications. The main challenge of data summarization is how to select *representative* data from a dataset. Sener and Savarese (2018) demonstrated that a dataset summary obtained by k-center clustering significantly reduces the time for learning process and yields a high inference accuracy. Here, assume that a biased summary is used as a training dataset for ML systems. Then, the ML models would provide biased or unfair results. This bias concern is nowadays widely known in the computer science community, e.g., a higher accuracy face recognition on specific groups (Buolamwini and Gebru, 2018), a tendency toward mis-labeling of some races in criminal recidivism tools (Angwin et al., 2016), and biased candidate selection in automated hiring systems (Tambe et al., 2019). For example, in the context of Web search result summarization, literature Kay et al. (2015) reported that Google Image, which provides a summary of matching images with a given query, can provide a summary wrongly dominated by some groups. Kleindessner et al. (2019) also reported that the fairness of a summarization result obtained by an algorithm without considering fairness can be arbitrarily bad.

As a *fair* data summarization that avoids the above biased results, the problem of fair k-center clustering has been studied. The set of centers returned by this problem is considered as a summary of a given dataset. Let P be a set of n points in a metric space, and P can be partitioned into disjoint subsets $P_1, ..., P_m$, where m is the number of demographic groups. Given a number k_i of centers for a group i, it is fair if the problem returns k_i centers for each group i. If we set $k_i = \frac{|P_i|}{|P|}k$, the set of centers preserves the proportion of each group. This *group fairness* has been widely employed in many existing works (not only for k-center clustering but also for other problems), e.g., (Angelidakis et al., 2022; Celis et al., 2018; Chiplunkar et al., 2020; Jones et al., 2020; Kleindessner et al., 2019; Moumoulidou et al., 2021; Thejaswi et al., 2022; Hajiaghayi et al., 2010; Krishnaswamy et al., 2011; Thejaswi et al., 2022, 2021), due to its effectiveness. Kleindessner et al. (2019) provides a case study (image summarization) of fair k-center clustering and shows its efficacy.

Because this fair k-center clustering problem is NP-hard, approximation algorithms were proposed in (Chiplunkar et al., 2020; Jones et al., 2020; Kleindessner et al., 2019). Jones et al. (2020) yielded an $O(nk)$ time 3-approximation algorithm. This is a nice re-

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

sult if P does not contain any *outliers*, but real-world datasets usually contain outliers (Amagata et al., 2021; Amagata and Hara, 2021; Amagata, 2022; Amagata et al., 2022; Amagata and Hara, 2024; Ceccarello et al., 2019; Ding et al., 2019; Gupta et al., 2017; Im et al., 2020; Matthew McCutchen and Khuller, 2008). Because the algorithms proposed in (Jones et al., 2020; Kleindessner et al., 2019) are based on Gonzalez's algorithm (Gonzalez, 1985), which iteratively retrieves the *furthest* point, they are sensitive to outliers (Amagata, 2023). If summaries contain outliers, their subsequent ML applications may face learning difficulty (Awasthi et al., 2017; Diakonikolas et al., 2020; Guruswami and Raghavendra, 2009). The above observations suggest that considering both fairness and outliers at the same time is important. However, this problem has not been well studied, and theoretically and empirically efficient algorithms for this problem have not been developed. This fact motivates us to address the problem of *fair k-center clustering with outliers.*

1.1 Preliminary

We use $dist(\cdot,\cdot)$ to denote the distance function that satisfies the triangle inequality. Given a point p and a subset $P' \subseteq P$, we define $dist(p, P') = \min_{p' \in P'} dist(p, p')$. For ease of presentation, we first define vanilla (unfair) k-center clustering with outliers (Bhaskara et al., 2019; Charikar et al., 2001).

Definition 1 ((k, z)-center clustering). *Given P, k, and $z < n$, this problem finds S^* s.t.*

$$S^* = \underset{S \subseteq P \setminus P_{out}, |S| = k}{\arg\min} \max_{p \in P \setminus P_{out}} dist(p, S), \tag{1}$$

where $P_{out} \subset P$ is a set of at most z points.

Outliers are z points with the furthest distance to their nearest centers. Similar to (Alon et al., 2003; Charikar et al., 2003; Ding et al., 2019; Huang et al., 2018; Li and Guo, 2018), we relax the above definition by allowing more points to be removed as z is not strict.

Definition 2 ($(k, (1+\epsilon)z)$-center clustering). *Given P, k, $z < n$, and $\epsilon > 0$, this problem is to find S^* that satisfies Equation (1) by redefining P_{out} as a set of at most $(1+\epsilon)z$ points.*

We incorporate a fairness constraint into the $(k, (1+\epsilon)z)$-center clustering problem. Each $p \in P$ belongs to one of m (demographic) groups, so $P = \bigcup_m P_i$, where P_i is a set of points belonging to group i. To effectively summarize P (or $P \setminus P_{out}$), k centers are selected, so it is desirable that we can specify the number k_i of centers for each group i. By combining this constraint and Definition 2, our problem is defined as follows:

Definition 3 (Fair $(k, (1+\epsilon)z)$-center clustering). *Given P, k, $\{k_1, ..., k_m\}$ such that $k \leq \sum_{i=1}^{m} k_i$, $z < n$, and $\epsilon > 0$, this problem is to find S^* such that*

$$S^* = \underset{S \subseteq P \setminus P_{out}, |S| = k, \forall i \in [1,m], |S \cap P_i| \leq k_i}{\arg\min} \max_{p \in P \setminus P_{out}} dist(p, S),$$

where $P_{out} \subset P$ is a set of at most $(1+\epsilon)z$ points.

This definition allows k and k_i to be specified independently, and "$|S \cap P_i| \leq k_i$" supports the case where $|P_i| < k_i$. This constraint follows (Chiplunkar et al., 2020; Jones et al., 2020).

We use $r^*_{k,z,m}$ to denote $\max_{p \in P \setminus P_{out}} dist(p, S^*)$ of the fair (k, z)-center clustering problem. Hence, $r^*_{k,z,1}$ means $\max_{p \in P \setminus P_{out}} dist(p, S^*)$ of the vanilla (k, z)-center clustering problem. In addition, we use $C^*_1, ..., C^*_k$ $(c^*_1, ..., c^*_k)$ to denote the optimal cluster (centers) of the fair (k, z)-center clustering problem. (C^*_i has c^*_i and points $\notin P_{out}$ whose nearest center is c^*_i). For a set S of centers, if $\max_{p \in P \setminus P_{out}} dist(p, S) \leq \lambda r^*_{k,z,m}$, where P_{out} follows Definition 3, we say that S is an λ-approximation result.

1.2 Combination of Known Techniques Cannot Have Theoretical Bound

Our problem is more challenging than the vanilla $(k, (1+\epsilon)z)$-center clustering problem, as we have an additional constraint. An intuitive approach that solves our problem is to combine an algorithm for the vanilla $(k, (1+\epsilon)z)$-center clustering, e.g., (Ding et al., 2019), with an approach that selects centers only from groups which do not violate the fairness constraint (Jones et al., 2020).

Specifically, this algorithm first randomly samples one point $p \in P$ and initializes an answer set S by setting $S = \{p\}$. Then, in the vanilla (k, z)-center clustering problem, this algorithm considers a set

$$F = \{p \,|\, p \in P, dist(p, S) > 2r^*_{k,z,1}\}, \tag{2}$$

where S is an intermediate result of the problem. We see that F is a set of points being sufficiently far from S. For the fair $(k, (1+\epsilon)z)$-center clustering problem, this algorithm "heuristically" considers a set

$$G = \bigcup_m \{p \,|\, p \in P_i, dist(p, S) > 2r^*_{k,z,m}, |\{p \cup S\} \cap P_i| \leq k_i\}. \tag{3}$$

The algorithm samples one point from G and adds it into S. This is repeated until $|S| = k$. Unfortunately, this approach loses any quality guarantees.

Claim 1. *The clustering quality of the above algorithm can be arbitrarily bad.*

All missing proofs appear in our supplementary file.

1.3 Our Contribution

Theoretical result. As shown above, simply combining existing techniques does not sound for our problem. We hence design a new algorithm that does not violate the fairness constraint while guaranteeing bounded clustering quality, success probability, and linear running time. Our result is as follows:

Theorem 1. *Given a set P of n points, k, $\{k_1, ..., k_m\}$, $z < n$, $\epsilon > 0$, and $\gamma > 0$ (an error factor of guessing $r^*_{k,z,m}$), there is an $O(nk)$ time algorithm that needs $O(n)$ space and yields a $(3+\gamma)$-approximation result for the fair $(k,(1+\epsilon)z)$-center clustering problem with probability at least $(1-\frac{z}{n})(\frac{\epsilon}{1+\epsilon})^{k-1}$, by returning exactly k centers and removing at most $(1+\epsilon)z$ points.*

In addition to the above new finding, we show that (i) improving the success probability cannot be done efficiently in Proposition 3 and (ii) random sampling yields a faster practical time while keeping a similar theoretical performance guarantee.

Efficiency of satisfying the fairness constraint. We show that the fairness constraint can be satisfied in $O(km\sqrt{k})$ time. This improves the state-of-the-art techniques (Chiplunkar et al., 2020; Jones et al., 2020). Specifically, Jones et al. (2020) and Chiplunkar et al. (2020) respectively need $O(n) + O(n\sqrt{k}\log k) + O(n(k + \log k)) + O(nk)$ and $O(nk + k^2m)$ time to satisfy the fairness constraint, whereas ours is independent of n.

Empirical performance. We empirically investigate the practical performance of our algorithm. Our experimental results demonstrate the efficiency and effectiveness of our algorithm.

2 Related Work

Fair k-center clustering without outliers. Our problem is a generalization of (Chen et al., 2016; Chiplunkar et al., 2020; Jones et al., 2020; Kleindessner et al., 2019). Kleindessner et al. (2019) considered the fair k-center clustering problem without outliers, and proposed a $(3\cdot 2^{m-1}-1)$-approximation algorithm that runs in $O(km^2n + km^4)$ time. Chen et al. (2016) regarded this fair k-center clustering without outliers problem as the matroid center problem, and they proposed a 3-approximation algorithm that runs in $\Omega(n^2 \log n)$ time.

Jones et al. (2020) improved these results by proposing a 3-approximation algorithm that runs in $O(nk)$ time. This is a theoretically interesting result, but its fairness satisfaction approach is still costly in practice. Our work improves this issue while considering the presence of outliers. In addition, our empirical study shows that, when P contains outliers, the algorithm proposed in (Jones et al., 2020) provides a large $\max_{p \in P \setminus P_{out}} dist(p, S)$ and its running time is longer than ours. Chiplunkar et al. (2020) also considered the problem of fair k-center clustering without outliers and proposed an $O(nk + mk^2)$ time streaming algorithm that yields almost 3-approximation. Distributed algorithms for fair k-center clustering were also designed in (Chiplunkar et al., 2020; Gan et al., 2023; Yuan et al., 2021). This setting is different from ours.

Problems with different objectives but the same constraint. Private k-center clustering (which has a constraint such that each cluster must have at least the specified number of members) (Angelidakis et al., 2022), k-means clustering (Hajiaghayi et al., 2010; Krishnaswamy et al., 2011; Thejaswi et al., 2022, 2021), and Max-Min diversification (Moumoulidou et al., 2021) also employ the same fairness constraint of our work. The objective functions of these problems are, however, different from ours, so their algorithms are not applicable to our problem.

Problem of k-clustering with different fairness definition. There are many fairness definitions even only in clustering problems. For example, Backurs et al. (2019) and Chierichetti et al. (2017) assume that there exist only two groups, and considered a constraint such that each cluster has a balanced group distribution. This fairness definition was generalized so that any number of groups is available. The works in (Ahmadian et al., 2019; Bera et al., 2019, 2022; Esmaeili et al., 2021; Dai et al., 2022) considered that each cluster should not be dominated by points with a specific group. Moreover, the works in (Bandyapadhyay et al., 2019; Almanza et al., 2022; Anegg et al., 2020; Jia et al., 2020) proposed k-clustering with outliers where z_i outliers are selected from each group i. Recently, Hotegni et al. (2023) have proposed fair range clustering which makes the number of clusters of each group variable.

The vanilla (k, z)-center clustering problem was studied extensively in (Bhaskara et al., 2019; Charikar et al., 2001; Ding et al., 2019; Matthew McCutchen and Khuller, 2008), since outliers are a standard observation in real-world data. Matthew McCutchen and Khuller (2008) designed a $(4+\gamma)$-approximation algorithm, and a 3-approximation algorithm was proposed in (Charikar et al., 2001). Recently, Bhaskara et al. (2019) and Ding et al. (2019) proposed $\tilde{O}(nk)$ time 2-approximation algorithms, where $\tilde{O}(\cdot)$ hides any polylog factor. As competitors, we use two state-of-the-art algorithms (Bhaskara et al., 2019; Ding et al., 2019) by incorporating the fairness satisfaction heuristic introduced in Section 1.2.

3 Our Algorithm

We assume that we can have r, a guess of $r^*_{k,z,m}$, such that $r \geq r^*_{k,z,m}$. This guess is done offline, similar to (Bhaskara et al., 2019; Chiplunkar et al., 2020). By using r as one of the inputs, our algorithm (Algorithm 1) selects $k = \sum k_i$ centers. Our algorithm builds on the vanilla (k, z)-clustering algorithm (Bhaskara et al., 2019). We extend this algorithm so that it can deal with the vanilla $(k, (1+\epsilon)z)$-clustering problem and can prepare center replacements to satisfy the fairness constraint. In a nutshell, our algorithm has two steps: a vanilla $(k, (1+\epsilon)z)$-clustering step (lines 1–22) and a fairness satisfaction step (lines 23–29).

The first step obtains at most k centers that do not take fairness into account. Specifically, we first initialize our solution S by a random point in P. Then, S is iteratively updated by a point p' randomly sampled from a set T of points p with $dist(p, S) > 2r$ when $|T| > (1+\epsilon)z$. At the same time, we build a conceptually cluster-group bipartite graph, i.e., there is an edge between a cluster and a group i iff the cluster contains at least one point $\in P_i$. In Algorithm 1, the vertices representing a cluster center and a group i are denoted by p' and v_i, respectively. Also, s and t represent start and termination vertices, respectively. The weight (capacity) of the edge between v_i and t is k_i, suggesting that group i can have *at most* k_i centers. The weight of the edge between p' and v_i is 1, and that of the edge between p' and s is also 1.

Note that if we run a state-of-the-art vanilla (k, z)-clustering algorithm (Ding et al., 2019) or the original algorithm of (Bhaskara et al., 2019) in this step, we cannot obtain the error-bounded result and performance guarantee. In addition, our approach in this step is essentially different from those of fair k-center algorithms (Chiplunkar et al., 2020; Jones et al., 2020; Kleindessner et al., 2019). Thanks to the above operation, we do not need to scan P after this step, and the time to satisfy the fairness is independent of n, see Proposition 1.

The second step is triggered iff $|S| < k$ or S violates the fairness constraint, i.e., there is a group i in S such that the number of centers in P_i is more than k_i. In this case, we obtain $k' \leq k$ cluster-group pairs that do not violate the fairness constraint via maximum bipartite matching (or solving the max-flow problem). Based on these pairs, we replace some centers in S to satisfy the fairness constraint. If $k' < k$, we choose $(k - k')$ centers arbitrarily so that the fairness constraint is not violated. Finally, we return S.

Algorithm 1: FAIR $(k, (1+\epsilon)z)$-CENTER CLUSTERING

```
Input: $P$, $k$, $\{k_1, k_2, ..., k_m\}$, $z$, $\epsilon$, and $r$ (a guess of $r^*_{k,z,m}$)
1: $S \leftarrow \{p'\}$                                ▷ $p'$ is a random point in $P$
2: $V \leftarrow \{s, t, p', v_1, ..., v_m\}$            ▷ a set of vertices
3: $E \leftarrow \{(s, p', 1), (v_1, t, k_1), ..., (v_m, t, k_m)\}$   ▷ a set of edges
4: while $|S| < k$ do
5:     $T \leftarrow \varnothing$
6:     for each $p \in P$ do
7:         if $\min_{p' \in S} dist(p, p') > 2r$ then
8:             $T \leftarrow T \cup \{p\}$
9:         if $p'$ is the nearest center of $p$ then
10:            Add $p$ into the cluster of $p'$
11:            Remove $p$ from the cluster of $p''$, where $p''$ is the previous nearest center of $p$
12:            if $p'$ newly has group $i$ then
13:                $E \leftarrow E \cup \{(p', v_i, 1)\}$
14:            if $p''$ loses group $i$ then
15:                $E \leftarrow E \backslash \{(p'', v_i, 1)\}$
16:    if $|T| > (1+\epsilon)z$ then
17:        $p' \leftarrow$ a randomly chosen point in $T$
18:        $S \leftarrow S \cup \{p'\}$, $V \leftarrow V \cup \{p'\}$
19:    else
20:        break
21: if $|S| = k$ then
22:    Run lines 6–15 without dealing with $T$
23: if $|S| < k$ or $S$ violates the fairness constraint then
24:    $\{(a, i), ..., (b, j)\} \leftarrow$ a max-flow algorithm on $G = (V, E)$
25:    for each $(x, y) \in \{(a, i), ..., (b, j)\}$ do
26:        Let $c_x$ be the center obtained at the $a$-th iteration in lines 2–18
27:        $c_x \leftarrow p$ such that $p \in P_y$, $p$ exists in the cluster of $c_x$, and $dist(c_x, p) \leq r$
28:    if $|\{(a, i), ..., (b, j)\}| = k' < k$ then
29:        Choose $(k - k')$ centers arbitrarily so that the fairness constraint is not violated
30: return $S$
```

3.1 Fairness Satisfaction and Time/Space Cost

Sections 3.1 and 3.2 address a non-trivial task of analyzing the performance of Algorithm 1, i.e., proving Theorem 1. First, we have the following results.

LEMMA 1. *Algorithm 1 always returns k centers without violating the fairness constraint.*

Proof. (i) The maximum bipartite matching (or max-flow) problem returns $k' \le k$ cluster-group pairs that do not violate the fairness constraint, because the capacity of edge (v_i, t), where t is the termination (or sink) vertex, is k_i, see line 3 of Algorithm 1. (That is, "at most" k_i centers are selected from group i.) (ii) If the maximum bipartite matching problem returns $k' < k$ cluster-group pairs, there must exist at least a group i with "less than" k_i centers. Hence, the remaining $(k - k')$ centers are arbitrarily chosen from such groups so that they do not violate the fairness constraint. □

Proposition 1. *Algorithm 1 needs $O(nk)$ time.*

Proof. Let us focus on the first step. The first iteration needs $O(1) + O(m) = O(m)$ time, because it is dominated by initializing m edges. We next consider the j-th iteration, where $j \in [2, k]$. Given a point in P, updating T needs $O(1)$ time. The edge update cost incurred in lines 12–15 can be done in $O(1)$ time by using an inverted file that is a set of postings lists where their keys are groups. Specifically, each center p' maintains its cluster members by postings lists $P'(i)$ that return cluster members belonging to P_i. That is, if p' is the nearest center of p and $p \in P_i$, p is maintained by $P'(i)$. This structure is based on a hash table, thus its amortized update time is $O(1)$, meaning that evaluating lines 12 and 14 needs $O(1)$ time. It is now easy to see that the j-th iteration needs $n \times O(1) = O(n)$ time. Then, the first step needs $O(m)+(k-1)O(n) = O(nk)$ time.

Next, we analyze the second step. By using Dinic's max-flow algorithm that runs in $O(\sqrt{|V|}|E|)$ time, we can obtain at most k cluster-group pairs in $O(\sqrt{k}km)$ time, since $|V| \le k + m + 2$, $|E| \le km$, and $m \le k$. Cluster center replacement based on the pairs (and choosing arbitrary $(k - k')$ centers) can be done in $O(k)$ time, through the inverted file structure.

For a given P, $m = O(1)$, thereby the total time is $O(nk) + O(\sqrt{k}k) = O(nk)$. □

Proposition 2. *Algorithm 1 needs $O(n + km)$ space.*

Proof. Algorithm 1 maintains T whose size is at most n, a bipartite graph with at most km edges, and the cluster members. Clearly, the total number of members is n, so this proposition holds. □

3.2 Approximation Bound and Success Probability

We here provide our approximation bound to prove Theorem 1. We introduce a lemma that derives our success probability. Hereinafter, we call non-outlier points inliers.

Lemma 2. *Focus on the first step of Algorithm 1. At its first iteration, the probability of sampling an inlier is $1 - \frac{z}{n}$. At its t-th iteration, where $t \ge 2$, the probability of sampling an inlier is at least $\frac{\epsilon}{1+\epsilon}$.*

We use c_j to denote the center selected at the j-th iteration of the first step of Algorithm 1.

Fact 1. *Assume that S contains only inliers. In this case, for any two centers $c_j, c_{j'} \in S$, they respectively belong to C^*_l and $C^*_{l'}$ such that $l \ne l'$.*

Let r be an λ-approximation of $r^*_{k,z,m}$ for $\lambda \ge 1$, i.e., $r \le \lambda r^*_{k,z,m}$. We prove that the first step of Algorithm 1 probabilistically guarantees 2λ-approximation of the *vanilla* $(k, (1+\epsilon)z)$-center clustering problem.

Lemma 3. *Given r such that $r^*_{k,z,m} \le r \le \lambda r^*_{k,z,m}$, the first step of Algorithm 1 yields a 2λ-approximation result for the vanilla $(k,(1+\epsilon)z)$-center clustering problem with probability at least $(1 - \frac{z}{n})(\frac{\epsilon}{1+\epsilon})^{k-1}$.*

Proof. Assume that S returned by the first step of Algorithm 1 contains only inliers, and this happens with probability at least $(1-\frac{z}{n})(\frac{\epsilon}{1+\epsilon})^{k-1}$, which is seen from Lemma 2. We have two cases in the first step: $|S| = k$ or $|S| < k$. Consider the former case. Fact 1 suggests that all centers in S belong to different C^*. From the triangle inequality, all points in $P \backslash P_{out}$ can be covered by k balls centered at the points in S with radius $2r^*_{k,z,m}$. Next, consider the latter case. This case means that Algorithm 1 has $|T| \le (1+\epsilon)z$ at the t-th iteration, where $t < k$. In this case, every point $p \in P \backslash T$ has $dist(p, S) \le 2r$. That is, by removing the points in T, the other points can be covered by $(t-1)$ balls centered at the points in S with radius $2r$, and the remaining $(k - t + 1)$ centers can be added into S arbitrarily. To summarize, in both cases, at least $n - (1+\epsilon)z$ points can be covered by exactly k balls centered at the points in S with radius $2r$. □

We next consider the fairness constraint and introduce an important lemma that derives Theorem 1.

Lemma 4. *Given r such that $r^*_{k,z,m} \le r \le \lambda r^*_{k,z,m}$, Algorithm 1 yields a 3λ-approximate solution, with probability at least $(1 - \frac{z}{n})(\frac{\epsilon}{1+\epsilon})^{k-1}$.*

Proof. Assume again that S returned by the first step of Algorithm 1 contains only inliers. Furthermore, assume that a center $c_j \in S$ selected by Algorithm 1 belongs to C^*_l in the optimal solution. From Fact 1, c^*_l belongs to the cluster of c_j in Algorithm 1. If $c^*_l \in P_i$, E certainly has edge (c_j, v_i). This observation suggests that there certainly exist feasible cluster-group pairs for the centers in S, and we can obtain them via the maximum bipartite graph matching. Therefore, we can assume that the cluster of c_j must have a center belonging to P_i as a result of the matching. In the cluster of c_j, there exists at least one point p that belongs to P_i and has $dist(c_j, p) \leq r$, because $dist(c_j, c_l^*) \leq r$. Hence, if we replace c_j with such a point, the radius of this cluster can be extended at most r from the triangle inequality. Putting this and Lemma 3 together, it is now easy to see that a 3λ-approximate solution is obtained when $|S| = k$. Even when $|S| < k$, the center replacements based on the matching result trivially do not violate the fairness constraint. The remaining centers can be arbitrarily chosen from $P \backslash T$ so that the fairness constraint is not violated. This case also has a 3λ-approximate solution from Lemma 3, so Lemma 4 holds. □

From Lemma 4, it is straightforward that, for r such that $r^*_{k,z,m} \leq r \leq (1+\gamma) r^*_{k,z,m}$, Algorithm 1 returns a $(3+\gamma')$-approximate solution with probability at least $(1 - \frac{z}{n})(\frac{\epsilon}{1+\epsilon})^{k-1}$, where $\gamma' = \gamma/3$. Now it is clear that Theorem 1 holds.

3.3 Discussion

Justification. Although our success probability has an exponent factor, it is necessary to satisfy the approximation bound, the fairness constraint, the bounded number of removed points, and the linear time at the same time, as justified below.

Proposition 3. *For any S such that $|S| \leq k$ and S contains at least one outlier, it is not guaranteed that (i) we have the worst-case approximation bound of $r^*_{k,z,m}$ and (ii) we can fix S so that it certainly satisfies $|S| = k$, the fairness constraint, and a guaranteed approximation bound of $r^*_{k,z,m}$ in polynomial time if P $\neq$ NP.*

Even the state-of-the-art *vanilla* (k, z)-center clustering algorithm (Ding et al., 2019) has a similar result w.r.t. success probability. Notice that Theorem 1 and Proposition 3 suggest what is possible (Theorem 1) and what is impossible (Proposition 3) w.r.t. the relationship between running time and success probability.

W next show that the success probability can be improved by sacrificing the time complexity a bit.

Guessing $r^*_{k,z,m}$. Boole's inequality derives the number β of trials necessary to guarantee returning a 3λ-approximate result at least once with *constant probability*, as discussed later. Now notice:

Fact 2. *We have $r^*_{k,z,1} \leq r^*_{k,z,m} \leq r^*_{k,0,m}$.*

This provides a lower-bound and an upper-bound of $r^*_{k,z,m}$, and this observation enables guessing $r^*_{k,z,m}$. (These bounds can be easily obtained offline by existing algorithms for the corresponding problems.) By running Algorithm 1 with $r = r^*_{k,z,1}, (1+\gamma) r^*_{k,z,1}, (1+\gamma)^2 r^*_{k,z,1}, ..., r^*_{k,0,m}$, we can have $r = (1+\gamma) r^*_{k,z,m}$ for some $\gamma > 0$. If $r^*_{k,0,m}/r^*_{k,z,1} = \text{poly}(n)$, this guessing needs $O(\frac{\beta}{\gamma} \log n) = \tilde{O}(1)$ trials (as γ, ϵ, and k are fixed during the trials).

Obtaining a $(3+\gamma')$-approximation result with constant probability. Recall that our algorithm returns a 3λ-approximate solution with probability at least $(1 - \frac{z}{n})(\frac{\epsilon}{1+\epsilon})^{k-1}$ by using r such that $r^*_{k,z,m} \leq r \leq \lambda r^*_{k,z,m}$. From Boole's inequality, we can obtain a 3λ-approximate solution with constant probability by repeating our algorithm $\beta = O(\frac{n}{n-z}(\frac{1+\epsilon}{\epsilon})^{k-1}) \approx O(1)$ times. (Therefore, we can obtain a $(3+\gamma')$-approximation result with constant probability in $O(nk(\frac{1+\epsilon}{\epsilon})^{k-1})$ time.) This assumes that $z \ll n$, $k = O(1)$, and $\epsilon = O(1)$, as we do not vary k and ϵ in the trials. For each trial of a radius $r \in [r^*_{k,z,1}, r^*_{k,0,m}]$, we run our algorithm β times. If $r \geq r^*_{k,z,m}$, we obtain a 3λ-approximate solution with constant probability. Therefore, with constant probability, we can obtain a $(3+\gamma')$-approximate result that is the best solution among the ones obtained by the trials of $r \geq r^*_{k,z,m}$.

Without repeating β times, even when $r \geq r^*_{k,z,m}$, we may fail to obtain a 3λ-approximate solution and the success probability decreases exponentially, i.e., the success probability becomes (approximately) $(1 - \frac{z}{n})(\frac{\epsilon}{1+\epsilon})^{k-1}$ to the power of $O(\log n)$. This suggests that the $(k, (1+\epsilon)z)$-center clustering problem has a trade-off relationship between time and success probability.

Optimizing practical performance through random sampling. Proposition 1 suggests that the main bottleneck of our algorithm is its first step. We minimize this cost through random sampling, and we introduce a corollary derived from Theorem 3 of (Charikar et al., 2003).

Corollary 1. *Let X be a set of $O(\frac{nk \ln n}{(1+\epsilon)^2 z})$ points randomly sampled from P. If S is a λ-approximate solution of $(k, (1+\epsilon)z)$-clustering on X, S is a λ-approximate solution of $(k, (1+\epsilon)^2 z)$-clustering on P with probability at least $1 - 2/n^2$.*

From this corollary, we extend Algorithm 1 to accelerate its first step. Note that existing fair k-center clustering works (Chiplunkar et al., 2020; Kleindessner et al., 2019; Jones et al., 2020) do not consider efficiency improvement by data size reduction.

In the extended first step, we randomly sample $O(\frac{nk \ln n}{(1+\epsilon)^2 z})$ points from P, select at most k centers via iterations, and then build an inverted file. This extension yields a slight difference compared with our main

Table 1: Summary of real datasets

Dataset	n	d	m	Description (A set of ...)
Adult-gender	48,843	6	2	U.S.A. census data (gender is used as group)
Adult-race	48,843	6	7	U.S.A. census data (race is used as group)
Covertype	580,812	10	7	cartographic variables (cover type is used as group)
Diabetes-gender	99,293	7	3	diabetes patient records (gender is used as group)
Diabetes–race	99,293	7	5	diabetes patient records (race is used as group)
KDD-Cup	311,029	38	16	packet records (attack type is used as group)
Mirai	764,137	115	2	packet records (attack/normal is used as group)

result, i.e., it achieves a bi-criteria approximation[1]. Due to the inverted file building, which is necessary to satisfy the fairness constraint, the $O(nk)$ time of Algorithm 1 remains. However, random samples can derive the following practical merits. (i) They reduce the time incurred by the iterations in the first step. (ii) As opposed to the theoretical result, they improve the practical success probability, because random samples contain fewer outliers than P.

4 Experiments

All experiments were conducted on a Ubuntu machine equipped with Xeon Platinum 8268 CPU@2.90GHz and 768GB RAM. Codes are available at an anonymous GitHub repository[2].

Competitors. We compared our algorithms[3] with the following competitors that return k centers and do not violate the fairness constraint. (These competitors have no theoretical approximation bound and success probability in our problem.)

(i) *ESA19* (Ding et al., 2019): this is an $O(nk\log(1+\epsilon)z)$ time algorithm based on Gonzalez's technique for the vanilla $(k,(1+\epsilon)z)$-center clustering problem. We incorporated the heuristic considered in (Jones et al., 2020; Kleindessner et al., 2019) into ESA19 (the one introduced in Section 1.2).

(ii) *NeurIPS19* (Bhaskara et al., 2019): this is an $O(nk)$ time algorithm for the vanilla (k,z)-center clustering problem. We also incorporated the same heuristic into this algorithm to satisfy the fairness constraint.

(iii) *ICML20-Jones* (Jones et al., 2020) and (iv) *Streaming* (Chiplunkar et al., 2020): these are $O(nk)$ time algorithms for the fair k-center clustering problem that does not consider outliers.

After we ran these algorithms, we removed $(1+\epsilon)z$ points with the furthest distance to their nearest clusters regardless of the theoretical guarantee. To be fair, all algorithms were implemented in C++, compiled by g++ 9.3.0 with O3 optimization, and single threaded.

Datasets. We used seven real datasets (Adult-gender, Adult-race, Covertype, Diabetes-gender, Diabetes-race, KDD-Cup, and Mirai) that are usually used as benchmark of the k-center clustering. They are publicly available at UCI Machine Learning Repository[4], and their summary is depicted in Table 1, where d represents the dimensionality. For diabetes, we removed records with missing values, and for Covertype and KDD-Cup, we removed binary attributes. After this, we injected z outliers into a given dataset so that they exist outside the data space uniformly at random. We used Euclidean distance as metric. The above setting follows those in (Bhaskara et al., 2019; Ceccarello et al., 2019; Ding et al., 2019; Im et al., 2020).

Result. We set $k=100$, $z=200$, and $\epsilon=9$ by default. In addition, as with (Jones et al., 2020), we set $k_i=\lceil\frac{|P_i|}{n}k\rceil$ so that the centers have the same proportionality as the entire dataset. We ran each algorithm 100 times by varying random seeds, and Tables 2 and 3 show maximum radius (i.e., $\max_{p\in P\setminus P_{out}} dist(p,S)$) and running time results, respectively. Ours-RS represents our algorithm with random sampling.

Table 2 shows that our algorithms provide better clustering results (i.e., the maximum radius, which is sometimes called cost, is minimized) than the other algorithms. In addition, from the standard deviation result, our algorithm with random sampling is more stable than the others. When m is relatively large (e.g., KDD-Cup), all algorithms tend to have a high standard deviation. Our algorithm with random sampling is still robust against m, as it provides a higher clustering result than the others on KDD-Cup. It is also important to notice that ICML20-Jones, which

[1]The number of removed points is at most $(1+\epsilon)^2z$ to satisfy a $(3+\gamma')$-approximate result theoretically (and the success probability has an additional $1-2/n^2\approx 1$ factor).

[2]https://github.com/amgt-d1/Fair-k-center-w-outliers

[3]We obtained r, a guess of $r^*_{k,z,m}$, offline as with (Bhaskara et al., 2019; Chiplunkar et al., 2020; Im et al., 2020), and we set $\gamma=0.1$.

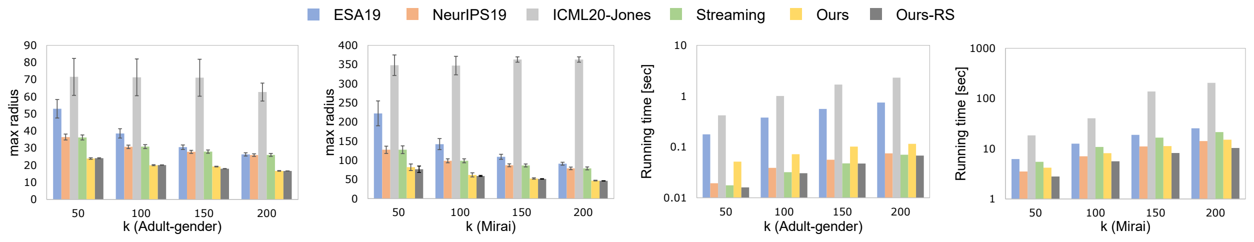
[4]https://archive.ics.uci.edu/ml/index.php

Table 2: Comparison of mean maximum radius (standard deviation), where $k = 100$, $k_i = \lceil \frac{|P_i|}{n} k \rceil$, $z = 200$, and $\epsilon = 9$.

Dataset	ESA19	NeurIPS19	ICML20-Jones	Streaming	Ours	Ours-RS
Adult-gender	38.59 (2.72)	30.76 (1.03)	71.37 (10.76)	30.89 (1.16)	**20.03** (0.31)	20.16 (**0.12**)
Adult-race	37.63 (2.20)	30.65 (0.88)	71.45 (10.98)	30.68 (1.04)	**20.00** (0.31)	20.12 (**0.10**)
Covertype	61.74 (2.38)	70.54 (1.79)	120.91 (9.43)	70.53 (3.14)	43.26 (0.79)	**42.62 (0.33)**
Diabetes-gender	42.99 (2.68)	39.89 (1.21)	104.54 (13.48)	40.29 (1.84)	**25.24** (0.32)	25.32 (**0.08**)
Diabetes-race	35.65 (2.77)	39.80 (1.19)	104.55 (13.44)	40.18 (1.84)	25.34 (0.32)	**25.27 (0.09)**
KDD-Cup	84.62 (10.38)	96.18 (9.66)	240.64 (18.60)	96.32 (6.36)	69.52 (13.61)	**40.04 (2.60)**
Mirai	142.16 (14.42)	98.89 (5.13)	347.29 (24.18)	98.87 (5.04)	62.02 (5.58)	**59.47 (2.05)**

Table 3: Comparison of average running time [sec], where $k = 100$, $k_i = \lceil \frac{|P_i|}{n} k \rceil$, $z = 200$, and $\epsilon = 9$.

Dataset	ESA19	NeurIPS19	ICML20-Jones	Streaming	Ours	Ours-RS
Adult-gender	0.38	0.04	1.01	**0.03**	0.07	**0.03**
Adult-race	0.34	0.04	0.87	0.05	0.05	**0.03**
Covertype	3.03	0.67	10.95	1.01	1.23	**0.57**
Diabetes-gender	0.63	**0.07**	1.85	0.08	0.16	**0.07**
Diabetes-race	0.59	**0.07**	1.66	0.08	0.14	0.08
KDD-Cup	1.81	0.46	3.59	1.17	0.53	**0.14**
Mirai	12.60	7.10	40.55	10.85	8.16	**4.27**



(a) Adult-gender (radius) (b) Mirai (radius) (c) Adult-gender (time) (d) Mirai (time)

Figure 1: Impact of k on maximum radius and running time

retrieves the furthest point in each iteration, yields a worse result than the others. This suggests that simply adding the furthest point into S is no more effective when P contains outliers.

Table 3 shows that our algorithm with random sampling is usually faster than the others. Although ICML20-Jones runs in $O(nk)$ time, it is the slowest, because it incurs a large time to satisfy the fairness constraint, as seen in Table 4 (which appears in our supplementary file).

Impact of k. We study the impact of k, and we here show the results on two datasets due to space limitation (those on the remaining datasets are similar). The tendency of the above results does not change even if k varies, which is seen from Figure 1 (notice that y-axis of Figures 1(c)–1(d) is log-scale).

Impacts of z and ϵ are reported in our supplementary file. In a nutshell, our algorithms are robust against z and ϵ, obtaining better (or at least competitive) performances than (with) the competitors.

Also, we investigated the impacts of m and n by using synthetic datasets. These results are reported in our supplementary file.

5 Conclusion

This paper addressed the problem of fair k-center clustering with outliers. For this problem, we proposed an $O(nk)$ time algorithm that achieves an almost 3-approximation bound with probabilistic guarantee. Our experimental results show that our algorithm is not only theoretically sound but also empirically effective and efficient.

Acknowledgements

This work was partially supported by AIP Acceleration Research JPMJCR23U2 and Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE) JPMJAP2328, JST.

References

Ahmadian, S., Epasto, A., Kumar, R., and Mahdian, M. (2019). Clustering without over-representation. In *KDD*, pages 267–275.

Almanza, M., Epasto, A., Panconesi, A., and Re, G. (2022). k-clustering with fair outliers. In *WSDM*, pages 5–15.

Alon, N., Dar, S., Parnas, M., and Ron, D. (2003). Testing of clustering. *SIAM Journal on Discrete Mathematics*, 16(3):393–417.

Amagata, D. (2022). Scalable and accurate density-peaks clustering on fully dynamic data. In *IEEE Big Data*, pages 445–454.

Amagata, D. (2023). Diversity maximization in the presence of outliers. In *AAAI*, volume 37, pages 12338–12345.

Amagata, D. and Hara, T. (2021). Fast density-peaks clustering: multicore-based parallelization approach. In *SIGMOD*, pages 49–61.

Amagata, D. and Hara, T. (2024). Efficient density-peaks clustering algorithms on static and dynamic data in euclidean space. *ACM Transactions on Knowledge Discovery from Data*, 18(1):1–27.

Amagata, D., Onizuka, M., and Hara, T. (2021). Fast and exact outlier detection in metric spaces: a proximity graph-based approach. In *SIGMOD*, pages 36–48.

Amagata, D., Onizuka, M., and Hara, T. (2022). Fast, exact, and parallel-friendly outlier detection algorithms with proximity graph in metric spaces. *The VLDB Journal*, 31(4):797–821.

Anegg, G., Angelidakis, H., Kurpisz, A., and Zenklusen, R. (2020). A technique for obtaining true approximations for k-center with covering constraints. In *IPCO*, pages 52–65.

Angelidakis, H., Kurpisz, A., Sering, L., and Zenklusen, R. (2022). Fair and fast k-center clustering for data summarization. In *ICML*, pages 669–702.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias. In *Ethics of Data and Analytics*, pages 254–264.

Awasthi, P., Balcan, M. F., and Long, P. M. (2017). The power of localization for efficiently learning linear separators with noise. *Journal of the ACM*, 63(6):1–27.

Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., and Wagner, T. (2019). Scalable fair clustering. In *ICML*, pages 405–413. PMLR.

Bandyapadhyay, S., Inamdar, T., Pai, S., and Varadarajan, K. (2019). A constant approximation for colorful k-center. In *ESA*, pages 12:1–12:14.

Bera, S., Chakrabarty, D., Flores, N., and Negahbani, M. (2019). Fair algorithms for clustering. *NeurIPS*, 32.

Bera, S. K., Das, S., Galhotra, S., and Kale, S. S. (2022). Fair k-center clustering in mapreduce and streaming settings. In *Web Conference*, pages 1414–1422.

Bhaskara, A., Vadgama, S., and Xu, H. (2019). Greedy sampling for approximate clustering in the presence of outliers. *NeurIPS*, 32.

Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAccT*, pages 77–91.

Ceccarello, M., Pietracaprina, A., and Pucci, G. (2019). Solving k-center clustering (with outliers) in mapreduce and streaming, almost as accurately as sequentially. *PVLDB*, 12(7):766–778.

Celis, E., Keswani, V., Straszak, D., Deshpande, A., Kathuria, T., and Vishnoi, N. (2018). Fair and diverse dpp-based data summarization. In *ICML*, pages 716–725.

Charikar, M., Khuller, S., Mount, D. M., and Narasimhan, G. (2001). Algorithms for facility location problems with outliers. In *SODA*, pages 642–651.

Charikar, M., O'Callaghan, L., and Panigrahy, R. (2003). Better streaming algorithms for clustering problems. In *STOC*, pages 30–39.

Chen, D. Z., Li, J., Liang, H., and Wang, H. (2016). Matroid and knapsack center problems. *Algorithmica*, 75(1):27–52.

Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. (2017). Fair clustering through fairlets. *NIPS*, 30.

Chiplunkar, A., Kale, S., and Ramamoorthy, S. N. (2020). How to solve fair k-center in massive data models. In *ICML*, pages 1877–1886.

Dai, Z., Makarychev, Y., and Vakilian, A. (2022). Fair representation clustering with several protected classes. In *FAccT*, pages 814–823.

Diakonikolas, I., Kane, D. M., and Manurangsi, P. (2020). The complexity of adversarially robust proper learning of halfspaces with agnostic noise. *NeurIPS*, 33:20449–20461.

Ding, H., Yu, H., and Wang, Z. (2019). Greedy strategy works for k-center clustering with outliers and coreset construction. In *ESA*.

Esmaeili, S., Brubach, B., Srinivasan, A., and Dickerson, J. (2021). Fair clustering under a bounded cost. *NeurIPS*, 34:14345–14357.

Gan, J., Golin, M., Yang, Z., and Zhang, Y. (2023). Fair k-center: a coreset approach in low dimensions. *arXiv preprint arXiv:2302.09911*.

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

Gupta, S., Kumar, R., Lu, K., Moseley, B., and Vassilvitskii, S. (2017). Local search methods for k-means with outliers. *PVLDB*, 10(7):757–768.

Guruswami, V. and Raghavendra, P. (2009). Hardness of learning halfspaces with noise. *SIAM Journal on Computing*, 39(2):742–765.

Hajiaghayi, M., Khandekar, R., and Kortsarz, G. (2010). Budgeted red-blue median and its generalizations. In *ESA*, pages 314–325.

Hotegni, S. S., Mahabadi, S., and Vakilian, A. (2023). Approximation algorithms for fair range clustering. In *ICML*, pages 13270–13284.

Huang, L., Jiang, S. H.-C., Li, J., and Wu, X. (2018). Epsilon-coresets for clustering (with outliers) in doubling metrics. In *FOCS*, pages 814–825.

Im, S., Qaem, M. M., Moseley, B., Sun, X., and Zhou, R. (2020). Fast noise removal for k-means clustering. In *AISTATS*, pages 456–466.

Jia, X., Sheth, K., and Svensson, O. (2020). Fair colorful k-center clustering. In *IPCO*, pages 209–222.

Jones, M., Nguyen, H., and Nguyen, T. (2020). Fair k-centers via maximum matching. In *ICML*, pages 4940–4949.

Kay, M., Matuszek, C., and Munson, S. A. (2015). Unequal representation and gender stereotypes in image search results for occupations. In *CHI*, pages 3819–3828.

Kleindessner, M., Awasthi, P., and Morgenstern, J. (2019). Fair k-center clustering for data summarization. In *ICML*, pages 3448–3457.

Krishnaswamy, R., Kumar, A., Nagarajan, V., Sabharwal, Y., and Saha, B. (2011). The matroid median problem. In *SODA*, pages 1117–1130.

Li, S. and Guo, X. (2018). Distributed k-clustering for data with heavy noise. *NeurIPS*, 31.

Matthew McCutchen, R. and Khuller, S. (2008). Streaming algorithms for k-center clustering with outliers and with anonymity. In *APPROX*, pages 165–178.

Moumoulidou, Z., McGregor, A., and Meliou, A. (2021). Diverse data selection under fairness constraints. In *ICDT*.

Sener, O. and Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach. In *ICLR*.

Tambe, P., Cappelli, P., and Yakubovich, V. (2019). Artificial intelligence in human resources management: Challenges and a path forward. *California Management Review*, 61(4):15–42.

Thejaswi, S., Gadekar, A., Ordozgoiti, B., and Osadnik, M. (2022). Clustering with fair-center representation: Parameterized approximation algorithms and heuristics. In *KDD*, pages 1749–1759.

Thejaswi, S., Ordozgoiti, B., and Gionis, A. (2021). Diversity-aware k-median: Clustering with fair center representation. In *ECMLPKDD*, pages 765–780.

Yuan, F., Diao, L., Du, D., and Liu, L. (2021). Distributed fair k-center clustering problems with outliers. In *PDCAT*, pages 430–440.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Not Applicable]

  (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
  (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A MISSING PROOFS

A.1 Proof of Claim 1

We use the same approach of proof of Lemma 1 in Bhaskara et al. (2019). Let α_t be the number of times that one of the outliers is added into S in the first t iterations. Furthermore, let n_t be the number of clusters such that $C_i^* \cap S_t \neq \varnothing$. We consider

$$\eta_t = \frac{\alpha_t |F_t \cap P_{in}|}{n_t}, \tag{4}$$

where P_{in} and F_t are respectively $P \backslash P_{out}$ and F in Equation (2) at the t-th iteration. Consider sampling a point at the $(t+1)$-th iteration. With probability $\frac{|C_i^* \cap G_t|}{|G_t|}$, where G_t is G in Equation (3) at the t-th iteration, we can sample an inlier, n_t decreases by one, and α_t keeps the same. On the other hand, with probability $1 - \frac{|C_i^* \cap G_t|}{|G_t|}$, we sample an outlier, n_t keeps the same, and α_t increases by one. Let $f_i = |C_i^* \cap F_t|$ and $g_i = |C_i^* \cap G_t|$. Then,

$$\mathbb{E}[\eta_{t+1}] \leq \sum_{i=1}^{k} \frac{g_i}{|G_t|} \frac{\alpha_t(|F_t \cap P_{in}| - f_i)}{n_t - 1} + (1 - \sum_{i=1}^{k} \frac{g_i}{|G_t|}) \frac{(\alpha_t + 1)|F_t \cap P_{in}|}{n_t}$$

Let $f = \sum_{i=1}^{k} f_i$ and $g = \sum_{i=1}^{k} g_i$. Since $G_t \subseteq F_t$,

$$\begin{aligned} \mathbb{E}[\eta_{t+1}] &\leq \sum_{i=1}^{k} \frac{g_i}{|G_t|} \frac{\alpha_t(|F_t \cap P_{in}| - f_i)}{n_t - 1} + (1 - \frac{g}{|G_t|}) \frac{\alpha_t f + f}{n_t} \\ &= \frac{\alpha_t}{|G_t|(n_t - 1)} (f^2 - \sum_{i=1}^{k} f_i^2) + (1 - \frac{g}{|G_t|}) \frac{\alpha_t f + f}{n_t} \end{aligned}$$

Because $f_i \leq n_t$, we have $\sum_i f_i^2 \geq \frac{f^2}{n_t}$ (Bhaskara et al., 2019). Therefore,

$$\mathbb{E}[\eta_{t+1}] \leq \frac{\alpha_t f^2}{|G_t| n_t} + (1 - \frac{g}{|G_t|}) \frac{\alpha_t f + f}{n_t} \leq \eta_t + (1 - \frac{g}{|G_t|}) \frac{f}{n_t}.$$

Because $1 - \frac{g}{|G_t|} \leq \frac{z}{|G_t|}$ and $f \leq |F_t|$,

$$\mathbb{E}[\eta_{t+1}] \leq \eta_t + \frac{z}{|G_t|} \frac{|F_t|}{n_t}.$$

By using the facts: (i) $\mathbb{E}[\eta_k] = \sum_{t=0}^{k-1} \mathbb{E}[\eta_{t+1} - \eta_t]$ and (ii) $n_t \geq k - t$,

$$\mathbb{E}[\eta_k] \leq \sum_{t=0}^{k-1} \frac{|F_t|}{|G_t|} \frac{z}{k - t}.$$

Because $\alpha_k = n_k$, we have the following result from Equation (4):

$$|F_t \cap P_{in}| \leq \sum_{t=0}^{k-1} \frac{|F_t|}{|G_t|} \frac{z}{k - t}.$$

It is easy to see that $|G_t|$ can be 0 in the worst case, thereby $|F_t \cap P_{in}|$ can be ∞ (or n). This suggests that (i) this heuristic cannot bound the number of removing points to satisfy a specific approximation bound w.r.t. radius (meaning that it needs to remove $O(n)$ points to satisfy 2-approximation bound) or (ii) has no approximation bound for covering points in $F_t \cap P_{in}$. $\square$

A.2 Proof of Lemma 2

As for the first iteration, it is straightforward. At the t-th iteration, where $t \geq 2$, we have at most z outliers, thus sampling an inlier can be done with probability at least $\frac{|T| - z}{|T|}$. Since $|T| > (1 + \epsilon)z$, we have $\frac{|T| - z}{|T|} \geq \frac{(1+\epsilon)z - z}{(1+\epsilon)z} = \frac{\epsilon}{1+\epsilon}$. $\square$

A.3 Proof of Fact 1

Recall that $r \geq r^*_{k,z,m}$ and T contains only points p with $dist(p, S) > 2r$. We have $dist(c_j, c_{j'}) > 2r$ for any two centers $c_j, c_{j'} \in S$. Assume that c_j belongs to C^*_l in the optimal case. From the triangle inequality, $c_{j'}$ cannot belong to C^*_l, which derives the above fact. $\square$

A.4 Proof of Proposition 3

Assume that S, which is returned by the first step of Algorithm 1, contains at least one outlier[5]. Furthermore, assume that $c_j \in S$ is an outlier. From the proof of Lemma 4, inlier centers certainly have cluster-group pairs that do not violate the fairness constraint. On the other hand, as c_j does not belong to any optimal clusters, the cluster of c_j may not have points in a specific group that is necessary to satisfy the fairness constraint. That is, there is a case where the maximum bipartite matching does not provide any group for the cluster of c_j. In this case, to satisfy the fairness constraint and a bounded approximation guarantee, c_j has to be replaced with an *inlier* in the specific group, say p, such that $dist(p, S) > 2r$. Finding p is, however, hard (or such points do not exist in T), since we cannot distinguish inliers from outliers in polynomial time if $\mathtt{P} \neq \mathtt{NP}$ (that is, this is to solve the k-center clustering with outliers, i.e., NP-hard problem, *exactly*), which results in case (ii). Then p should be an *arbitrary* point that satisfies the fairness constraint as in Algorithm 1, but this makes the radius of the cluster of p arbitrarily long in the worst case since p still may be an outlier and it does not belong to the cluster of c_j. This proves case (i). $\square$

A.5 Proof of Corollary 1

Let X be a set of $O(\frac{k \ln n}{\delta \epsilon'^2})$ points randomly sampled from P for $\epsilon' \in (0, 1)$ and $\delta \in (0, 1)$. Furthermore, let r' be the maximum radius derived from a set of k-centers obtained on X. Assume $r' = \lambda r^*_{k,z,1}$, then the set of k-centers provides $r' = \lambda r^*_{k,z,1}$ even on P by removing at most $(1 + \epsilon')^2 \delta n$ points with probability at least $1 - 2/n^2$ (Charikar et al., 2003). We now want $(1 + \epsilon')^2 \delta n = (1 + \epsilon)^2 z$. Then, $\delta = \frac{(1+\epsilon)^2 z}{(1+\epsilon')^2 n}$. By fixing ϵ', the sample size becomes $O(\frac{nk \ln n}{(1+\epsilon)^2 z})$. $\square$

B ADDITIONAL EXPERIMENTS

B.1 Impact of z

We report the results of our experiments with varying z in Figure 2. In terms of maximum radius, the results are similar to those of the experiments with varying k. As for running time, our algorithm with random sampling is the fastest or is competitive with the fastest algorithm. That is, our algorithm has the best trade-off between time and maximum radius.

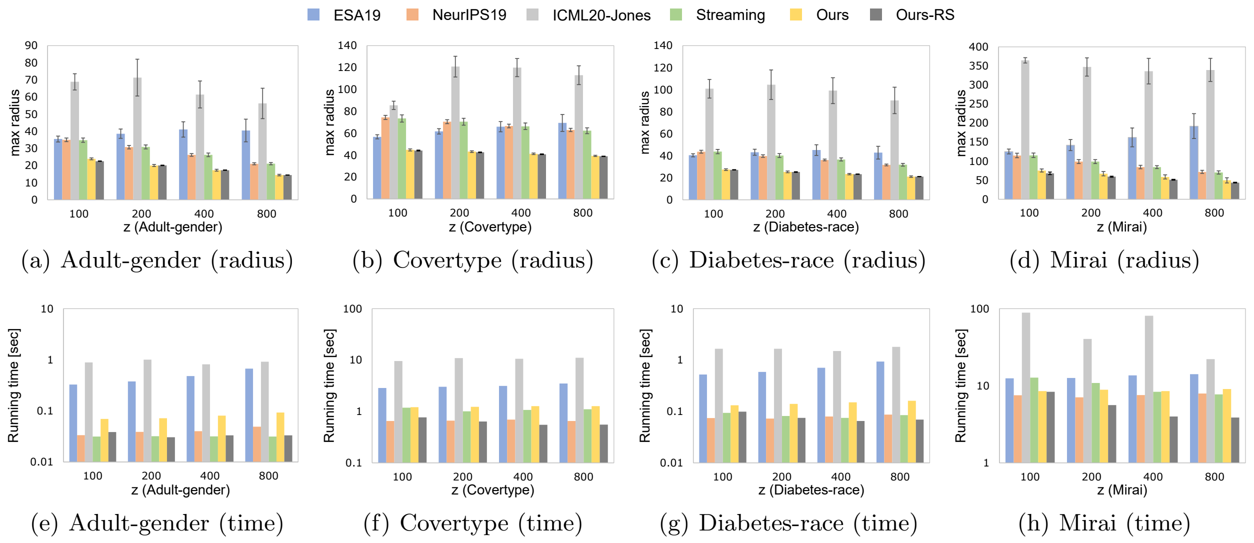
B.2 Impact of ϵ

Last, we show the results of our experiments with varying ϵ in Figure 3. They are similar to those in Figure 2. We observe that our algorithm is robust to k, z, and ϵ w.r.t. maximum radius and its practical running time is reasonable.

B.3 Group Fairness Matching Time

Table 4 shows that our fairness satisfaction approach is much faster than the existing techniques. We omit the matching time of our algorithm with random sampling, as it is essentially the same as the result of Ours.

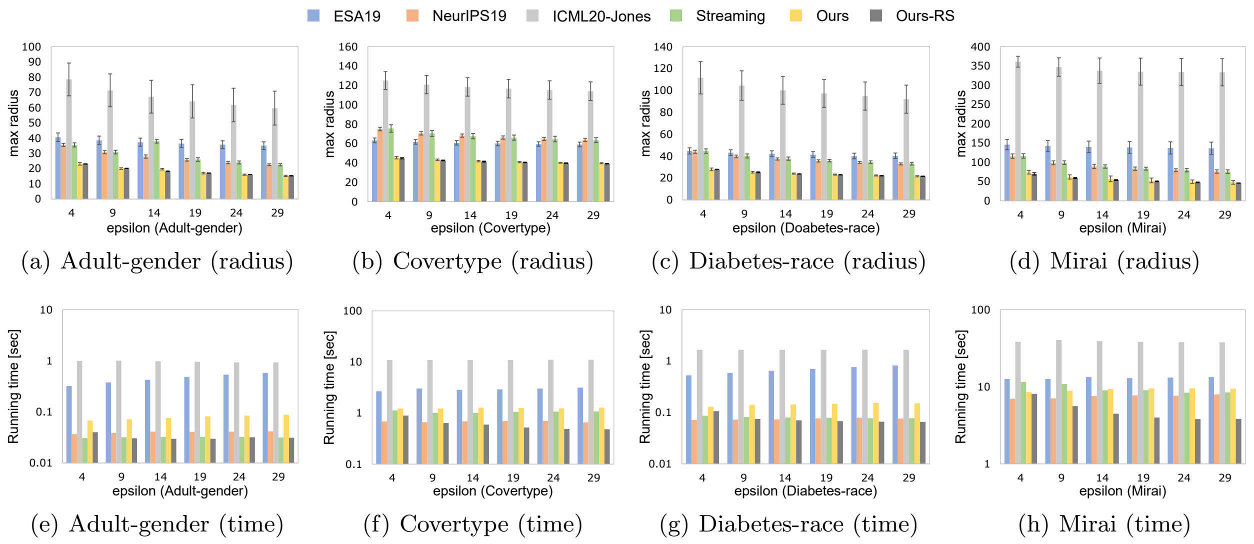
[5] In this case, with probability at least 3/4, S can cover $(n - 4z \ln k)$ points by using $|S|$ balls centered at the points in S with radius $2r$ (Bhaskara et al., 2019). Although this case has the approximation guarantee, the bounded number of removed points, and the success probability, we miss the fairness satisfaction, the main focus of our problem.



(a) Adult-gender (radius) (b) Covertype (radius) (c) Diabetes-race (radius) (d) Mirai (radius)

(e) Adult-gender (time) (f) Covertype (time) (g) Diabetes-race (time) (h) Mirai (time)

Figure 2: Impact of z on maximum radius and running time



(a) Adult-gender (radius) (b) Covertype (radius) (c) Diabetes-race (radius) (d) Mirai (radius)

(e) Adult-gender (time) (f) Covertype (time) (g) Diabetes-race (time) (h) Mirai (time)

Figure 3: Impact of ϵ on maximum radius and running time

Table 4: Comparison of matching time [millisec], where $k = 100$, $k_i = \lceil \frac{|P_i|}{n} k \rceil$, $z = 200$, and $\epsilon = 9$.

Dataset	ICML20-Jones	Streaming	Ours
Adult-gender	980.12	0.99	**0.01**
Adult-race	842.82	1.03	**0.01**
Covertype	10358.80	1.03	**0.02**
Diabetes-gender	1792.31	1.11	**0.01**
Diabetes-race	1605.02	1.00	**0.01**
KDD-Cup	3156.84	1.59	**0.04**
Mirai	33183.30	1.17	**0.02**

Table 5: Maximum radius and running time of our algorithm (with random sampling) with different m on synthetic dataset

m	10	20	30	40	50
Maximum radius	4.90 (5.13)	4.89 (5.14)	4.91 (5.11)	4.94 (5.20)	4.93 (5.20)
Time [sec]	0.13 (0.03)	0.13 (0.03)	0.13 (0.03)	0.13 (0.03)	0.13 (0.03)

Table 6: Maximum radius of our algorithm (with random sampling) with different n on synthetic dataset

n	100,000	250,000	500,000	750,000	1,000,000
Maximum radius	4.90 (5.13)	5.15 (5.37)	5.39 (5.64)	5.43 (5.77)	5.66 (5.84)

B.4 Result on Synthetic Datasets

Dataset. We next confirm the practical scalablity of our algorithm (with random sampling) w.r.t. m and n. To do this, we generated synthetic datasets that have 50 Gaussian clusters in a 5-dimensional space. We assigned each point with one group uniformly at random. Outliers were injected as with the real datasets case.

Impact of m. We set $n = 100,000$ and studied how the performance of our algorithm (with random sampling) varies by varying m ($k = 50$ and k_i was set in the same way as in the experiments on real datasets). We ran our algorithm (with random sampling) 100 times for each $m \in [10, 50]$. Recall that $m \leq k$ if all groups have to appear as centers.

Table 5 shows the maximum radius and running time of our algorithm (with random sampling) with different m. Our algorithm yields a stable result as its maximum radius does not vary with different m (under the same data distribution). This is a similar result as those on Adult and Diabetes. The running time result also does not vary.

Impact of n. We last confirm how the maximum radius yielded by our algorithm (with random sampling) varies with different n by using synthetic datasets. (Its running time is linear to n, as seen in Theorem 2.) We fixed $m = 10$, and Table 6 shows the result. As n increases, the maximum radius increases. This is reasonable: a larger n means that we have more points, thus covering them generally needs a larger radius. Nevertheless, the increment of the maximum radius is slight, see the cases where $n = 100,000$ and $n = 1,000,000$.