Greedy Strategy Works for k-Center Clustering with Outliers and Coreset Construction

Hu Ding
School of Computer Science and Technology,
University of Science and Technology of China, He Fei, China
huding@ustc.edu.cn

Haikuo Yu
School of Computer Science and Technology,
University of Science and Technology of China, He Fei, China
yhk7786@mail.ustc.edu.cn

Zixiu Wang
School of Computer Science and Technology,
University of Science and Technology of China, He Fei, China
wzx2014@mail.ustc.edu.cn

Abstract

We study the problem of k-center clustering with outliers in arbitrary metrics and Euclidean space. Though a number of methods have been developed in the past decades, it is still quite challenging to design quality guaranteed algorithm with low complexity for this problem. Our idea is inspired by the greedy method, Gonzalez's algorithm, for solving the problem of ordinary k-center clustering. Based on some novel observations, we show that this greedy strategy actually can handle k-center clustering with outliers efficiently, in terms of clustering quality and time complexity. We further show that the greedy approach yields small coreset for the problem in doubling metrics, so as to reduce the time complexity significantly. Our algorithms are easy to implement in practice. We test our method on both synthetic and real datasets. The experimental results suggest that our algorithms can achieve near optimal solutions and yield lower running times comparing with existing methods.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases k-center clustering, outliers, coreset, doubling metrics, random sampling

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.40

Funding *Hu Ding*: this work was supported in part by NSF through grant CCF-1656905.

Acknowledgements The authors want to thank the anonymous reviewers for their helpful comments and suggestions for improving the paper.

1 Introduction

Clustering is one of the most fundamental problems in data analysis [25]. Given a set of elements, the goal of clustering is to partition the set into several groups based on their similarities or dissimilarities. Several clustering models have been extensively studied, such as k-center, k-median, and k-means clustering [9]. In reality, datasets often are noisy and contain outliers. Moreover, outliers could seriously affect the final results in data analysis [14]. Clustering with outliers can be viewed as a generalization of ordinary clustering problems; however, the existence of outliers makes the problems to be much more challenging.

We focus on the problem of *k-center clustering with outliers* in this paper. Given a metric space with n vertices and a pre-specified number of outliers $z < n$, the problem is to find k balls to cover at least $n - z$ vertices and minimize the maximum radius of the balls. The problem also can be defined in Euclidean space so that the cluster centers can be any points in the space (i.e., not restricted to be selected from the input points). The

2-approximation algorithms for ordinary k-center clustering (without outliers) were given in [18, 22], and it was proved that any approximation ratio lower than 2 implies $P = NP$. A 3-approximation algorithm for k-center clustering with outliers in arbitrary metrics was proposed by Charikar et al. [15]; for the problem in Euclidean space, their approximation ratio becomes 4. A following streaming $(4+\epsilon)$-approximation algorithm was proposed by McCutchen and Khuller [33]. Recently, Chakrabarty et al. [13] proposed a 2-approximation algorithm for metric k-center clustering with outliers (but it is unclear of the resulting approximation ratio for the problem in Euclidean space). Existing algorithms often have high time complexities. For example, the complexities of the algorithms in [15,33] are $O(kn^2 \log n)$ and $O\big(\frac{1}{\epsilon}(kzn + (kz)^2 \log \Phi)\big)$ respectively, where Φ is the ratio of the optimal radius to the smallest pairwise distance among the vertices; the algorithm in [13] needs to solve a complicated model of linear programming and the exact time complexity is not provided. The coreset based idea of Badoiu et al. [7] needs to enumerate a large number of possible cases and also yields a high complexity. Several distributed algorithms for k-center clustering with outliers were proposed recently [12, 19, 30, 32]; most of these distributed algorithms, to our best knowledge, rely on the sequential algorithm [15].

In this paper, we aim to design quality guaranteed algorithm with low complexity for the problem of k-center clustering with outliers. Our idea is inspired by the greedy method from Gonzalez [18] for solving ordinary k-center clustering. Based on some novel insights, we show that this greedy method also works for the problem with outliers (Section 2). Our approach can achieve the approximation ratio 2 with respect to the clustering cost (i.e., the radius); moreover, the time complexity is linear in the input size. Charikar et al. [16] showed that if more than z outliers are allowed to remove, the random sampling technique can be applied to reduce the data size for metric k-center clustering with outliers. Recently, Huang et al. [23] showed a similar result for instances in Euclidean space (and they name the sample as "robust coreset"). In Section 2.3, we prove that the sample size of [23] can be further reduced.

We also consider the problem in doubling metrics, motivated by the fact that many real-world datasets often manifest low intrinsic dimensions [8]. For example, image sets usually can be represented in low dimensional manifold though the Euclidean dimension of the image vectors can be very high. "Doubling dimension" is widely used for measuring the intrinsic dimensions of datasets [35] (the formal definition is given in Section 1.1). Rather than assuming the whole (X, d) has a low doubling dimension, we only assume that **the inliers of the given data have a low doubling dimension $\rho > 0$.** We do not have any assumption on the outliers; namely, the outliers can scatter arbitrarily in the space. We believe that this assumption captures a large range of high dimensional instances in reality.

With the assumption, we show that our approach can further improve the clustering quality. In particular, the greedy approach is able to construct a coreset for the problem of k-center clustering with outliers; as a consequence, the time complexity can be significantly reduced if running existing algorithms on the coreset (Section 3). *Coreset* construction is a technique for reducing data size so as to speedup the algorithms for many optimization problems; we refer the reader to the surveys [5, 34] for more details. The size of our coreset is $2z + O\big((2/\mu)^\rho k\big)$, where μ is a small parameter measuring the quality of the coreset; the construction time is $O((\frac{2}{\mu})^\rho kn)$. Note that z and k are often much smaller than n in practice; the coefficient 2 of z actually can be further reduced to be arbitrarily close to 1, by increasing the coefficient of the second term $(2/\mu)^\rho k$. Moreover, our coreset is a natural "composable coreset" [24] which could be potentially applied to distributed clustering with outliers. Very recently, Ceccarello et al. [12] also provided a coreset for k-center clustering

with z outliers in doubling metrics, where their size is $T = O((k+z)(\frac{24}{\mu})^{\rho})$ with $O(nT)$ construction time. Thus our result is a significant improvement in terms of coreset size and construction time. Huang et al. [23] considered the coreset construction for k-median/means clustering with outliers in doubling metrics, however, their method cannot be extended to the case of k-center. Aghamolaei and Ghodsi [2] considered the coreset construction in doubling metrics for ordinary k-center clustering without outliers.

Our proposed algorithms are easy to implement in practice. To study the performance of our algorithms, we test them on both synthetic and real datasets in Section 4. The experimental results suggest that our method outperforms existing methods in terms of clustering quality and running time. Also, the running time can be significantly reduced via building coreset where the clustering quality can be well preserved simultaneously.

Due to the space limit, some details are omitted here, and we refer the reader to the full version of our paper.

1.1 Preliminaries

We consider the problem of k-center with outliers in arbitrary metrics and Euclidean space $\mathbb{R}^D$. Let (X, d) be a metric, where X contains n vertices and $d(\cdot,\cdot)$ is the distance function; with a slight abuse of notation, we also use the function d to denote the shortest distance between two subsets $X_1, X_2 \subseteq X$, i.e., $d(X_1, X_2) = \min_{p\in X_1, q\in X_2} d(p,q)$. We assume that the distance between any pair of vertices in X is given in advance; for the problem in Euclidean space, it takes $O(D)$ time to compute the distance between any pair of points. Below, we introduce several important definitions that are used throughout the paper.

▶ **Definition 1** (k-Center Clustering with Outliers). *Given a metric (X, d) with two positive integers k and $z < n$, k-center clustering with outliers is to find a subset $X' \subseteq X$, where $|X'| \geq n - z$, and k centers $\{c_1, \cdots, c_k\} \subseteq X$, such that $\max_{p\in X'} \min_{1\leq j\leq k} d(p, c_j)$ is minimized. If given a set P of n points in $\mathbb{R}^D$, the problem is to find a subset $P' \subseteq P$, where $|P'| \geq n - z$, and k centers $\{c_1, \cdots, c_k\} \subset \mathbb{R}^D$, such that $\max_{p\in P'} \min_{1\leq j\leq k} ||p - c_j||$ is minimized.*

Note. For the sake of convenience, we describe the following definitions only in terms of metric space. In fact, the definitions can be easily modified for the problem in Euclidean space.

In this paper, we always use X_{opt}, a subset of X with size $n - z$, to denote the subset yielding the optimal solution. Also, let $\{C_1, \cdots, C_k\}$ be the k clusters forming X_{opt}, and the resulting clustering cost be r_{opt}; that is, each C_j is covered by an individual ball with radius r_{opt}.

Usually, optimization problems with outliers are challenging to solve. Thus we often relax our goal and allow to miss a little more outliers in practice. Actually the same relaxation idea has been adopted by a number of works on clustering with outliers before [3, 16, 23, 30].

▶ **Definition 2** ($(k,z)_\epsilon$-Center Clustering). *Let (X, d) be an instance of k-center clustering with z outliers, and $\epsilon \geq 0$. $(k,z)_\epsilon$-center clustering is to find a subset X' of X, where $|X'| \geq n - (1+\epsilon)z$, such that the corresponding clustering cost of Definition 1 on X' is minimized.*

(i) *Given a set A of cluster centers ($|A|$ could be larger than k), the resulting clustering cost,*

$$\min\Big\{\max_{p\in X'}\min_{c\in A} d(p,c) \mid X' \subseteq X, |X'| \geq n - (1+\epsilon)z\Big\} \tag{1}$$

is denoted by $\phi_\epsilon(X, A)$.

(ii) *If* $|A| = k$ *and* $\phi_\epsilon(X, A) \leq \alpha r_{opt}$ *with* $\alpha > 0$[1]*, it is called an* α*-approximation. Moreover, if* $|A| = \beta k$ *with* $\beta > 1$*, it is called an* (α, β)*-approximation.*

Obviously, the problem in Definition 1 is a special case of $(k, z)_\epsilon$-center clustering with $\epsilon = 0$. Further, Definition 1 and 2 can be naturally extended to **weighted case:** each vertex p has a non-negative weight w_p and the total weight of outliers should be equal to z; the distance $d(p, c_j)$ in the objective function is replaced by $w_p \cdot d(p, c_j)$. Then we have the following definition of coreset.

▶ **Definition 3** (Coreset). *Given a small parameter* $\mu \in (0, 1)$ *and an instance* (X, d) *of* k*-center clustering with* z *outliers, a set* $S \subseteq X$ *is called a* μ*-coreset of* X*, if each vertex of* S *is assigned a non-negative weight and* $\phi_0(S, H) \in (1 \pm \mu)\phi_0(X, H)$ *for any set* $H \subseteq X$ *of* k *vertices.*

Given a large-scale instance (X, d), we can run existing algorithm on its coreset S to compute an approximate solution for X; if $|S| \ll n$, the resulting running time can be significantly reduced. Formally, we have the following claim.

▷ Claim 4. If the set H yields an α-approximation of the μ-coreset S, it yields an $\alpha \times \frac{1+\mu}{1-\mu}$-approximation of X.

As mentioned before, we also consider the case with low doubling dimension. Roughly speaking, doubling dimension describes the expansion rate of the metric. For any $p \in X$ and $r \geq 0$, we use $Ball(p, r)$ to denote the ball centered at p with radius r.

▶ **Definition 5** (Doubling Dimension). *The doubling dimension of a metric* (X, d) *is the smallest number* $\rho > 0$*, such that for any* $p \in X$ *and* $r \geq 0$*,* $X \cap Ball(p, 2r)$ *is always covered by the union of at most* 2^ρ *balls with radius* r*.*

2 Algorithms for $(k, z)_\epsilon$-Center Clustering

For the sake of completeness, let us briefly introduce the algorithm of [18] for ordinary k-center clustering first. Initially, it arbitrarily selects a vertex from X, and iteratively selects the following $k - 1$ vertices, where each j-th step $(2 \leq j \leq k)$ chooses the vertex having the largest minimum distance to the already selected $j - 1$ vertices; finally, each input vertex is assigned to its nearest neighbor of these selected k vertices. It can be proved that this greedy strategy results in a 2-approximation of k-center clustering; the algorithm also works for the problem in Euclidean space and results in the same approximation ratio. In this section, we show that a modified version of Gonzalez's algorithm yields approximate solutions for $(k, z)_\epsilon$-center clustering.

In Section 2.1 and 2.2, we present our results for metric k-center with outliers. Actually, it is easy to see that Algorithm 1 and 2 yield the same approximation ratios if the input instance is a set of points in Euclidean space (the analysis is almost identical, and we omit the details due to the space limit); only the running times are different, since it takes $O(D)$ time to compute distance between two points in $\mathbb{R}^D$.

[1] Since we remove more than z outliers, it is possible to have an approximation ratio $\alpha < 1$, i.e, $\phi_\epsilon(X, A) < r_{opt}$.

Algorithm 1 Bi-criteria Approximation Algorithm.

Input: An instance (X, d) of metric k-center clustering with z outliers, and $|X| = n$; parameters $\epsilon > 0$, $\eta \in (0, 1)$, and $t \in \mathbb{Z}^+$.

1. Let $\gamma = z/n$ and initialize a set $E = \emptyset$.
2. Initially, $j = 1$; randomly select $\frac{1}{1-\gamma} \log \frac{1}{\eta}$ vertices from X and add them to E.
3. Run the following steps until $j = t$:
   a. $j = j + 1$ and let Q_j be the farthest $(1 + \epsilon)z$ vertices of X to E (for each vertex $p \in X$, its distance to E is $\min_{q \in E} d(p, q)$).
   b. Randomly select $\frac{1+\epsilon}{\epsilon} \log \frac{1}{\eta}$ vertices from Q_j and add them to E.

Output E.

2.1 $(2, O(\frac{1}{\epsilon}))$-Approximation

Here, we consider bi-criteria approximation that returns more than k cluster centers. The main challenge for implementing Gonzalez's algorithm is that the outliers and inliers are mixed in X; for example, the selected vertex, which has the largest minimum distance to the already selected vertices, is very likely to be an outlier, and therefore the clustering quality could be arbitrarily bad. Instead, our strategy is to take a small sample from the farthest subset. We implement our idea in Algorithm 1. For simplicity, let γ denote z/n in the algorithm; usually we can assume that γ is a value much smaller than 1. We prove the correctness of Algorithm 1 below.

▶ **Lemma 6.** *With probability at least $1 - \eta$, the set E in Step 2 of Algorithm 1 contains at least one point from X_{opt}.*

Since $|X_{opt}|/|X| = 1 - \gamma$, Lemma 6 can be easily obtained by the following folklore claim.

▷ Claim 7. Let U be a set of elements and $V \subseteq U$ with $\frac{|V|}{|U|} = \tau > 0$. Given $\eta \in (0, 1)$, if one randomly samples $\frac{1}{\tau} \log \frac{1}{\eta}$ elements from U, with probability at least $1 - \eta$, the sample contains at least one element from V.

Recall that $\{C_1, C_2, \cdots, C_k\}$ are the k clusters forming X_{opt}. Denote by $\lambda_j(E)$ the number of the clusters which have non-empty intersection with E at the beginning of j-th round in Step 3 of Algorithm 1. For example, initially $\lambda_1(E) \geq 1$ by Lemma 6. Obviously, if $\lambda_j(E) = k$, i.e., $C_l \cap E \neq \emptyset$ for any $1 \leq l \leq k$, E yields a 2-approximation for k-center clustering with outliers through the triangle inequality.

▷ Claim 8. If $\lambda_j(E) = k$, then $\phi_0(X, E) \leq 2r_{opt}$.

▶ **Lemma 9.** *In each round of Step 3 of Algorithm 1, with probability at least $1 - \eta$, either (1) $d(Q_j, E) \leq 2r_{opt}$ or (2) $\lambda_j(E) \geq \lambda_{j-1}(E) + 1$.*

Proof. Suppose that (1) is not true, i.e., $d(Q_j, E) > 2r_{opt}$, and we prove that (2) is true. Let $\mathcal{J}$ include all the indices $l \in \{1, 2, \cdots, k\}$ with $E \cap C_l \neq \emptyset$. We claim that $Q_j \cap C_l = \emptyset$ for each $l \in \mathcal{J}$. Otherwise, let $p \in Q_j \cap C_l$ and $p' \in E \cap C_l$; due to the triangle inequality, we know that $d(p, p') \leq 2r_{opt}$ which is in contradiction to the assumption $d(Q_j, E) > 2r_{opt}$. Thus, $Q_j \cap X_{opt}$ only contains the vertices from C_l with $l \notin \mathcal{J}$. Moreover, since the number of outliers is z, we know that $\frac{|Q_j \cap X_{opt}|}{|Q_j|} \geq \frac{\epsilon}{1+\epsilon}$. By Claim 7, if randomly selecting $\frac{1+\epsilon}{\epsilon} \log \frac{1}{\eta}$ vertices from Q_j, with probability at least $1 - \eta$, the sample contains at least one vertex from $Q_j \cap X_{opt}$; also, the vertex must come from $\cup_{l \notin \mathcal{J}} C_l$. That is, (2) $\lambda_j(E) \geq \lambda_{j-1}(E) + 1$ happens. ◀

If (1) of Lemma 9 happens, i.e., $d(Q_j, E) \leq 2r_{opt}$, then it implies that

$$\max_{p \in X \setminus Q_j} d(p, E) \leq 2r_{opt}; \tag{2}$$

moreover, since $|Q_j| = (1+\epsilon)z$, we have $\phi_\epsilon(X, E) \leq 2r_{opt}$. Next, we assume that (1) in Lemma 9 never happens, and prove that $\lambda_j(E) = k$ with constant probability when $j = \Theta(k)$. The following idea actually has been used by Aggarwal et al. [1] for achieving a bi-criteria approximation for k-means clustering. Define a random variable x_j: $x_j = 1$ if $\lambda_j(E) = \lambda_{j-1}(E)$, or 0 if $\lambda_j(E) \geq \lambda_{j-1}(E)+1$, for $j = 1, 2, \cdots$. So $\mathbb{E}[x_j] \leq \eta$ by Lemma 9 and

$$\sum_{1 \leq s \leq j} (1 - x_s) \leq \lambda_j(E). \tag{3}$$

Also, let $J_j = \sum_{1 \leq s \leq j}(x_s - \eta)$ and $J_0 = 0$. Then, $\{J_0, J_1, J_2, \cdots\}$ is a super-martingale with $J_{j+1} - J_j < 1$ (more details are shown in the full version of our paper). Through *Azuma-Hoeffding inequality* [4], we have $Pr(J_t \geq J_0 + \delta) \leq e^{-\frac{\delta^2}{2t}}$ for any $t \in \mathbb{Z}^+$ and $\delta > 0$. Let $t = \frac{k+\sqrt{k}}{1-\eta}$ and $\delta = \sqrt{k}$, the inequality implies that

$$Pr(\sum_{1 \leq s \leq t} (1 - x_s) \geq k) \geq 1 - e^{-\frac{1-\eta}{4}}. \tag{4}$$

Combining (3) and (4), we know that $\lambda_t(E) \geq k$ with probability at least $1 - e^{-\frac{1-\eta}{4}}$. Moreover, $\lambda_t(E) = k$ directly implies that E is a 2-approximate solution by Claim 8. Together with Lemma 6, we have the following theorem.

▶ **Theorem 10.** *Let $\epsilon > 0$. If we set $t = \frac{k+\sqrt{k}}{1-\eta}$ for Algorithm 1, with probability at least $(1-\eta)(1-e^{-\frac{1-\eta}{4}})$, $\phi_\epsilon(X, E) \leq 2r_{opt}$.*

Quality and Running time. If $\frac{1}{\eta}$ and $\frac{1}{1-\gamma}$ are constant numbers, $|E|$ will be $O(\frac{k}{\epsilon})$ and Theorem 10 implies that E is a $\big(2, O(\frac{1}{\epsilon})\big)$-approximation for $(k, z)_\epsilon$-center clustering of X with constant probability. In each round of Step 3, there are $O(\frac{1}{\epsilon})$ new vertices added to E, thus it takes $O(\frac{1}{\epsilon}n)$ time to update the distances from the vertices of X to E; to select the set Q_j, we can apply the linear time selection algorithm [10]. Overall, the running time of Algorithm 1 is $O(\frac{k}{\epsilon}n)$. If the given instance is in $\mathbb{R}^D$, the running time will be $O(\frac{k}{\epsilon}nD)$.

Further, we consider the instances under some practical assumption, and provide new analysis of Algorithm 1. In reality, the clusters are usually not too small, compared with the number of outliers. For example, it is rare to have a cluster C_l that $|C_l| \ll z$.

▶ **Theorem 11.** *If each optimal cluster C_l has size at least ϵz for $1 \leq l \leq k$, the set E of Algorithm 1 is a $\big(4, O(\frac{1}{\epsilon})\big)$-approximation for the problem of $(k, z)_0$-center clustering with constant probability.*

Compared with Theorem 10, Theorem 11 shows that we can exactly exclude z outliers (rather than $(1+\epsilon)z$), though the approximation ratio with respect to the radius becomes 4.

Proof of Theorem 11. We take a more careful analysis on the proof of Lemma 9. If (1) never happens, eventually $\lambda_j(E)$ will reach k and thus $\phi_0(X, E) \leq 2r_{opt}$ (Claim 8). So we focus on the case that (1) happens before $\lambda_j(E)$ reaching k. Suppose at j-th round, $d(Q_j, E) \leq 2r_{opt}$ but $\lambda_j(E) < k$. We consider two cases **(i)** there exists some $l_0 \notin \mathcal{J}$ such that $C_{l_0} \subseteq Q_j$ and **(ii)** otherwise.

Algorithm 2 2-Approximation Algorithm.

Input: An instance (X, d) of metric k-center clustering with z outliers, and $|X| = n$; a parameter $\epsilon > 0$.

1. Initialize a set $E = \emptyset$.
2. Let $j = 1$; randomly select one vertex from X and add it to E.
3. Run the following steps until $j = k$:
   a. $j = j + 1$ and let Q_j be the farthest $(1+\epsilon)z$ vertices to E.
   b. Randomly select one vertex from Q_j and add it to E.

Output E.

For **(i)**, we have $C_{l_0} \subseteq Q_j$ for some $l_0 \notin \mathcal{J}$. Note that we assume $|C_{l_0}| \geq \epsilon z$, i.e., $\frac{|C_{l_0}|}{|Q_j|} \geq \frac{\epsilon}{1+\epsilon}$. Using the same manner in the proof of Lemma 9, we know that (2) $\lambda_j(E) \geq \lambda_{j-1}(E)+1$ happens with probability $1-\eta$. Thus, if **(i)** is always true, we can continue Step 3 and eventually $\lambda_j(E)$ will reach k, that is, a $\left(2, O(\frac{1}{\epsilon})\right)$-approximation of $(k,z)_0$-center clustering is obtained with constant probability.

For **(ii)**, we have $C_l \setminus Q_j \neq \emptyset$ for all $l \notin \mathcal{J}$. Together with the assumption $d(Q_j, E) \leq 2r_{opt}$, we know that there exists $q_l \in C_l \setminus Q_j$ (for each $l \notin \mathcal{J}$) such that $d(q_l, E) \leq d(Q_j, E) \leq 2r_{opt}$. Consequently, we have that $\forall q \in C_l$,

$$d(q, E) \leq \quad ||q - q_l|| + d(q_l, E) \leq 4r_{opt} \text{ (see the left of Figure. 1).} \tag{5}$$

Note that for any $l \in \mathcal{J}$, $d(E, C_l) \leq 2r_{opt}$ by the triangle inequality. Thus,

$$\phi_0(X, E) \leq \quad \max_{q \in \cup_{l=1}^{k} C_l} d(q, E) \leq 4r_{opt}. \tag{6}$$

So a $\left(4, O(\frac{1}{\epsilon})\right)$-approximation of $(k,z)_0$-center clustering is obtained. ◀

2.2 2-Approximation

If k is a constant, we show that a single-criterion 2-approximation can be achieved. Actually, we use the same strategy as Section 2.1, but run only k rounds with each round sampling only one vertex. See Algorithm 2.

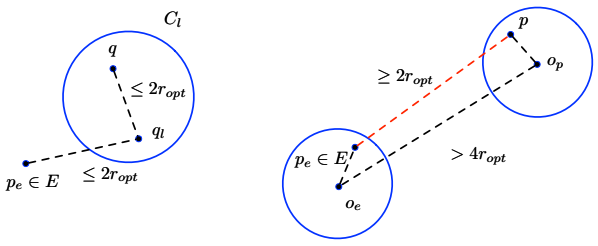

Figure 1 Left: p_e is a point of E having distance $\leq 2r_{opt}$ to p_l; right: p_e is any point of E, o_e and o_p are the centers taking charge of p_e and p.

Denote by $\{v_1, \cdots, v_k\}$ the k sampled vertices of E. Actually, the proof of Theorem 12 is similar to the analysis in Section 2.1. The only difference is that the probability that (2) $\lambda_j(E) \geq \lambda_{j-1}(E)+1$ happens is at least $\frac{\epsilon}{1+\epsilon}$. Also note that $v_1 \in X_{opt}$ with probability $1-\gamma$ ($\gamma = z/n$). If all of these events happen, either we obtain a 2-approximation before k steps

(i.e., $d(E, X \setminus Q_j) \leq 2r_{opt}$ for some $j < k$), or $\{v_1, \cdots, v_k\}$ fall into the k optimal clusters $C_1, C_2, \cdots, C_k$ separately (i.e., $\lambda_k(E) = k$). No matter which case happens, we always obtain a 2-approximation with respect to $(k, z)_\epsilon$-center clustering. So we have Theorem 12.

▶ **Theorem 12.** *Algorithm 2 returns a 2-approximation for the problem of $(k, z)_\epsilon$-center clustering on X, with probability at least $(1-\gamma)(\frac{\epsilon}{1+\epsilon})^{k-1}$. The running time is $O(kn)$. If the given instance is in $\mathbb{R}^D$, the running time will be $O(knD)$.*

To boost the probability of Theorem 12, we just need to repeatedly run the algorithm; the success probability is easy to calculate by taking the union bound.

▶ **Corollary 13.** *If we run Algorithm 2 $O\big(\frac{1}{1-\gamma}(\frac{1+\epsilon}{\epsilon})^{k-1}\big)$ times, with constant probability, at least one time the algorithm returns a 2-approximation for the problem of $(k, z)_\epsilon$-center clustering.*

Similar to Theorem 11, we consider the practical instances. We show that the quality of Theorem 12 can be preserved even exactly excluding z outliers, if the optimal clusters are "well separated". The property was also studied for other clustering problems in practice [17, 26]. Let $\{o_1, \cdots, o_k\}$ be the k cluster centers of the optimal clusters $\{C_1, \cdots, C_k\}$.

▶ **Theorem 14.** *Suppose that each optimal cluster C_l has size at least ϵz and $||o_l - o_{l'}|| > 4r_{opt}$ for $1 \leq l \neq l' \leq k$. Then with probability at least $(1-\gamma)(\frac{\epsilon}{1+\epsilon})^{k-1}$, Algorithm 2 returns a 2-approximation for the problem of $(k, z)_0$-center clustering.*

Proof. Initially, we know that $\lambda_1(E) = 1$ with probability $1 - \gamma$. Suppose that at the beginning of the j-th round of Algorithm 2 with $2 \leq j \leq k$, E already has $j - 1$ vertices separately falling in $j - 1$ optimal clusters; also, we still let $\mathcal{J}$ be the set of the indices of these $j - 1$ clusters. Then we have the following claim.

▷ Claim 15. $|Q_j \cap (\cup_{l \notin \mathcal{J}} C_l)| \geq \epsilon z$.

Proof. For any $p \in \cup_{l \notin \mathcal{J}} C_l$, we have

$$d(p, E) > 4r_{opt} - r_{opt} - r_{opt} = 2r_{opt} \tag{7}$$

from triangle inequality and the assumption $||o_l - o_{l'}|| > 4r_{opt}$ for $1 \leq l \neq l' \leq k$ (see the right of Figure. 1). In addition, for any $p \in \cup_{l \in \mathcal{J}} C_l$, we have

$$d(p, E) \leq 2r_{opt}. \tag{8}$$

We consider two cases. If $d(Q_j, E) \leq 2r_{opt}$ at the current round, then (7) directly implies that $\cup_{l \notin \mathcal{J}} C_l \subseteq Q_j$ (recall Q_j is the set of farthest vertices to E); thus $|Q_j \cap (\cup_{l \notin \mathcal{J}} C_l)| = |\cup_{l \notin \mathcal{J}} C_l| \geq \epsilon z$ by the assumption that any $|C_l| \geq \epsilon z$. Otherwise, $d(Q_j, E) > 2r_{opt}$. Then $Q_j \cap (\cup_{l \in \mathcal{J}} C_l) = \emptyset$ by (8). Moreover, since there are only z outliers and $|Q_j| = (1 + \epsilon)z$, we know that $|Q_j \cap (\cup_{l \notin \mathcal{J}} C_l)| \geq \epsilon z$. ◁

Claim 15 reveals that with probability at least $\frac{\epsilon}{1+\epsilon}$, the new added vertex falls in $\cup_{l \notin \mathcal{J}} C_l$, i.e., $\lambda_j(E) = \lambda_{j-1}(E) + 1$. Overall, we know that $\lambda_k(E) = k$, i.e., E is a 2-approximation of $(k, z)_0$-center clustering (by Claim 8), with probability at least $(1-\gamma)(\frac{\epsilon}{1+\epsilon})^{k-1}$. ◀

2.3 Reducing Data Size via Random Sampling

Given a metric (X, d), Charikar et al. [16] showed that we can use a random sample S to replace X. Recall $\gamma = z/n$. Let $|S| = O(\frac{k}{\epsilon^2\gamma} \ln n)$ and E be an α-approximate solution of $(k, z)_\epsilon$-center clustering on (S, d), then E is an α-approximate solution of $(k, z)_{O(\epsilon)}$-center clustering on (X, d) with constant probability. In D-dimensional Euclidean space, Huang et al. [23] showed a similar result, where the sample size $|S| = \tilde{O}(\frac{1}{\epsilon^2\gamma^2}kD)$[2] (to be consistent with our paper, we change the notations in their theorem). In this section, we show that the sample size of [23] can be further improved to be $\tilde{O}(\frac{1}{\epsilon^2\gamma}kD)$, which can be a significant improvement if $\frac{1}{\gamma} = \frac{n}{z}$ is large.

Let P be a set of n points in $\mathbb{R}^D$. Consider the range space $\Sigma = (P, \Pi)$ where each range $\pi \in \Pi$ is the complement of union of k balls in $\mathbb{R}^D$. We know that the VC dimension of balls is $O(D)$ [4], and therefore the VC dimension of union of k balls is $O(kD \log k)$ [11]. That is, the VC dimension of the range space Σ is $O(kD \log k)$. Let $\epsilon \in (0, 1)$, and an "ϵ-sample" S of P is defined as follows: $\forall \pi \in \Pi$, $\left|\frac{|\pi \cap P|}{|P|} - \frac{|\pi \cap S|}{|S|}\right| \leq \epsilon$; roughly speaking, S is an approximation of P with an additive error inside each range π. Given a range space with VC dimension m, an ϵ-sample can be easily obtained via uniform sampling [4], where the success probability is $1 - \lambda$ and the sample size is $O\big(\frac{1}{\epsilon^2}(m \log \frac{m}{\epsilon} + \log \frac{1}{\lambda})\big)$ for any $0 < \lambda < 1$. For our problem, we need to replace the "ϵ" of the "ϵ-sample" by $\epsilon\gamma$ to guarantee that the number of uncovered points is bounded by $\big(1 + O(\epsilon)\big)\gamma n$ (we show the details below); the resulting sample size will be $\tilde{O}(\frac{1}{\epsilon^2\gamma^2}kD)$ that is the same as the sample size of [23] (we assume that the term $\log \frac{1}{\lambda}$ is a constant for convenience).

Actually, the front factor $\frac{1}{\epsilon^2\gamma^2}$ of the sample size can be further reduced to be $\frac{1}{\epsilon^2\gamma}$ by a more careful analysis. We observe that there is no need to guarantee the additive error for each range π (as the definition of ϵ-sample). Instead, only a multiplicative error for the ranges covering at least γn points should be sufficient. Note that when a range covers more points, the multiplicative error is weaker than the additive error and thus the sample size is reduced. For this purpose, we use *relative approximation* [21, 31]: let $S \subseteq P$ be a subset of size $\tilde{O}(\frac{1}{\epsilon^2\gamma}kD)$ chosen uniformly at random, then with constant probability,

$$\forall \pi \in \Pi, \ \left|\frac{|\pi \cap P|}{|P|} - \frac{|\pi \cap S|}{|S|}\right| \leq \epsilon \times \max\left\{\frac{|\pi \cap P|}{|P|}, \gamma\right\}. \tag{9}$$

We formally state our result below.

▶ **Theorem 16.** *Let P be an instance for the problem of k-center clustering with outliers in $\mathbb{R}^D$ as described in Definition 1, and $S \subseteq P$ be a subset of size $\tilde{O}(\frac{1}{\epsilon^2\gamma}kD)$ chosen uniformly at random. Suppose $\epsilon \leq 0.5$. Let S be a new instance for the problem of k-center clustering with outliers where the number of outliers is set to be $z' = (1 + \epsilon)\gamma|S|$. If E is an α-approximate solution of $(k, z')_\epsilon$-center clustering on S, then E is an α-approximate solution of $(k, z)_{O(\epsilon)}$-center clustering on P, with constant probability.*

Proof. We assume that S is a relative approximation of P and (9) holds (this happens with constant probability). Let $\mathbb{B}_{opt}$ be the set of k balls covering $(1 - \gamma)n$ points induced by the optimal solution for P, and $\mathbb{B}_S$ be the set of k balls induced by an α-approximate solution of $(k, z')_\epsilon$-center clustering on S. Suppose the radius of each ball in $\mathbb{B}_{opt}$ (resp., $\mathbb{B}_S$) is r_{opt} (resp., r_S). We denote the complements of $\mathbb{B}_{opt}$ and $\mathbb{B}_S$ as π_{opt} and π_S, respectively.

[2] The asymptotic notation $\tilde{O}(f) = O\big(f \cdot polylog(\frac{kD}{\epsilon\gamma})\big)$.

First, since $\mathbb{B}_{opt}$ covers $(1-\gamma)n$ points of P and S is a relative approximation of P, we have

$$\frac{|\pi_{opt} \cap S|}{|S|} \leq \frac{|\pi_{opt} \cap P|}{|P|} + \epsilon \times \max\left\{\frac{|\pi_{opt} \cap P|}{|P|}, \gamma\right\} = (1+\epsilon)\gamma \tag{10}$$

by (9). That is, the set balls $\mathbb{B}_{opt}$ cover at least $\big(1-(1+\epsilon)\gamma\big)|S|$ points of S, and therefore it is a feasible solution for the instance S with respect to the problem of k-center clustering with z' outliers. Since $\mathbb{B}_S$ is an α-approximate solution of $(k,z')_\epsilon$-center clustering on S, we have

$$r_S \leq \alpha r_{\text{opt}}; \qquad |\pi_S \cap S| \leq (1+\epsilon)z' = (1+\epsilon)^2\gamma|S|. \tag{11}$$

Now, we claim that

$$|\pi_S \cap P| \leq \frac{(1+\epsilon)^2}{1-\epsilon}\gamma|P|. \tag{12}$$

Assume that (12) is not true, then (9) implies $\left|\frac{|\pi_S\cap P|}{|P|} - \frac{|\pi_S\cap S|}{|S|}\right| \leq \epsilon \times \max\left\{\frac{|\pi_S\cap P|}{|P|}, \gamma\right\} = \epsilon\frac{|\pi_S\cap P|}{|P|}$. So $\frac{|\pi_S\cap S|}{|S|} \geq (1-\epsilon)\frac{|\pi_S\cap P|}{|P|} > (1+\epsilon)^2\gamma$, which is in contradiction with the second inequality of (11), and thus (12) is true. We assume $\epsilon \leq 0.5$, so $\frac{1}{1-\epsilon} \leq 1+2\epsilon$ and $\frac{(1+\epsilon)^2}{1-\epsilon} = 1+O(\epsilon)$. Consequently (12) and the first inequality of (11) together imply that $\mathbb{B}_S$ is an α-approximate solution of $(k,z)_{O(\epsilon)}$-center clustering on P. ◀

3 Coreset Construction in Doubling Metrics

In this section, we always assume the following is true by default:

Given an instance (X,d) of k-center clustering with outliers, the metric (X_{opt},d), i.e., the metric formed by the set of inliers, has a constant doubling dimension $\rho > 0$.

We do not have any restriction on the outliers $X \setminus X_{opt}$. Thus the above assumption is more relaxed and practical than assuming the whole (X,d) has a constant doubling dimension. From Definition 5, we directly know that each optimal cluster C_l of X_{opt} can be covered by 2^ρ balls with radius $r_{opt}/2$ (see the left figure in Figure. 2). Imagine that the instance (X,d) has $2^\rho k$ clusters, where the optimal radius is at most $r_{opt}/2$. Therefore, we can just replace k by $2^\rho k$ when running Algorithm 1, so as to reduce the approximation ratio (i.e., the ratio of the resulting radius to r_{opt}) from 2 to 1.

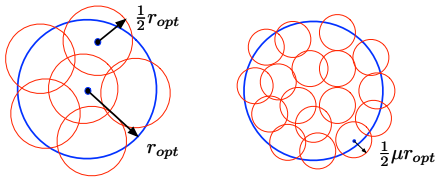

Figure 2 Illustrations for Theorem 17 and 18.

▶ **Theorem 17.** *If we set $t = \frac{2^\rho k + 2^{\rho/2}\sqrt{k}}{1-\eta}$ for Algorithm 1, with probability $(1-\eta)(1-e^{-\frac{1-\eta}{4}})$, $\phi_\epsilon(X,E) \leq r_{opt}$. So the set E is a $\big(1, O(\frac{2^\rho}{\epsilon})\big)$-approximation for the problem of $(k,z)_\epsilon$-center clustering, and the running time is $O(2^\rho \frac{k}{\epsilon} n)$.*

Algorithm 3 The Coreset Construction.

Input: An instance (X, d) of metric k-center clustering with z outliers, and $|X| = n$; parameters η and $\mu \in (0, 1)$.

1. Let $l = (\frac{2}{\mu})^\rho k$.
2. Set $\epsilon = 1$ and run Algorithm 1 $t = \frac{l+\sqrt{l}}{1-\eta}$ rounds. Denote by $\tilde{r}$ the maximum distance between E and X by excluding the farthest $2z$ vertices, after the final round of Algorithm 1.
3. Let $X_{\tilde{r}} = \{p \mid p \in X \text{ and } d(p, E) \leq \tilde{r}\}$.
4. For each vertex $p \in X_{\tilde{r}}$, assign it to its nearest neighbor in E; for each vertex $q \in E$, let its weight be the number of vertices assigning to it.
5. Add $X \setminus X_{\tilde{r}}$ to E; each vertex of $X \setminus X_{\tilde{r}}$ has weight 1.

Output E as the coreset.

If considering the problem in Euclidean space $\mathbb{R}^D$ where the doubling dimension of the inliers is ρ, the running time becomes $O(2^\rho \frac{k}{\epsilon} nD)$. Inspired by Theorem 17, we can further construct coreset for k-center clustering with outliers (see Definition 3). Let $\mu \in (0, 1)$, and for simplicity we assume that $\log 2/\mu$ is an integer. If applying Definition 5 recursively, we know that each C_l is covered by $2^{\rho \log 2/\mu} = (\frac{2}{\mu})^\rho$ balls with radius $\frac{\mu}{2} r_{opt}$, and X_{opt} is covered by $(\frac{2}{\mu})^\rho k$ such balls in total. See the right figure in Figure. 2. Based on this observation, we have Algorithm 3 for constructing μ-coreset.

▶ **Theorem 18.** *With constant probability, Algorithm 3 outputs a μ-coreset E of k-center clustering with z outliers. The size of E is at most $2z + O\big((\frac{2}{\mu})^\rho k\big)$, and the construction time is $O((\frac{2}{\mu})^\rho kn)$.*

Remark. **(1)** The previous ideas based on uniform sampling [16, 23] (also our idea in Section 2.3) cannot get rid of the violation on the number of outliers; the sample sizes will become infinity if not allowing to remove more than z outliers. Our coreset in Theorem 18 works for removing z outliers exactly. Consequently, our coreset can be used for existing algorithms of k-center clustering with outliers, such as [15], to reduce their complexities. **(2)** Another feature is that our coreset is a natural composable coreset. If X (or the point set P) is partitioned into L parts, we can run Algorithm 3 for each part, and obtain a coreset with size $\Big(2z + O\big((\frac{2}{\mu})^\rho k\big)\Big)L$ in total (the proof is almost identical to the proof of Theorem 18 below). So our coreset construction can potentially be applied to distributed clustering with outliers. **(3)** The coefficient 2 of z actually can be further reduced by modifying the value of ϵ in Step 2 of Algorithm 3 (we just set $\epsilon = 1$ for simplicity). In general, the size of E is $(1 + \epsilon)z + O\big(\frac{1}{\epsilon}(\frac{2}{\mu})^\rho k\big)$ and the construction time is $O(\frac{1}{\epsilon}(\frac{2}{\mu})^\rho kn)$ (or $O(\frac{1}{\epsilon}(\frac{2}{\mu})^\rho knD)$ in $\mathbb{R}^D$).

Proof of Theorem 18. Similar to Theorem 17, we know that $|X_{\tilde{r}}| = n - 2z$ and $\tilde{r} \leq 2 \times \frac{\mu}{2} r_{opt} = \mu r_{opt}$ with constant probability in Algorithm 3. Thus, the size of E is $|X \setminus X_{\tilde{r}}| + O\big((\frac{2}{\mu})^\rho k\big) = 2z + O\big((\frac{2}{\mu})^\rho k\big)$. Moreover, it is easy to see that the running time of Algorithm 3 is $O((\frac{2}{\mu})^\rho kn)$.

Next, we show that E is a μ-coreset of X. For each vertex $q \in E$, denote by $w(q)$ the weight of q; for the sake of convenience in our proof, we view each q as a set of $w(q)$ overlapping unit weight vertices. Thus, from the construction of E, we can see that there is a bijective mapping f between X and E, where

$$||p - f(p)|| \leq \tilde{r} \leq \mu r_{opt}, \qquad \forall p \in X. \tag{13}$$

Let $H = \{c_1, c_2, \cdots, c_k\}$ be any k vertices of X. Suppose that H induces k clusters $\{A_1, A_2, \cdots, A_k\}$ (resp., $\{B_1, B_2, \cdots, B_k\}$) with respect to the problem of k-center clustering with z outliers on E (resp., X), where each A_j (resp., B_j) has the cluster center c_j for $1 \leq j \leq k$. Let $r_E = \phi_0(E, H)$ and $r_X = \phi_0(X, H)$, respectively. Also, let r'_E (resp., r'_X) be the smallest value r, such that for any $1 \leq j \leq k$, $f(B_j) \subseteq Ball(c_j, r)$ (resp., $f^{-1}(A_j) \subseteq Ball(c_j, r)$). We need the following claim.

▷ Claim 19. $|r'_E - r_X| \leq \mu r_{opt}$ and $|r'_X - r_E| \leq \mu r_{opt}$.

In addition, since $\{f(B_1), \cdots, f(B_k)\}$ also form k clusters for the instance E with the fixed k cluster centers of H, we know that $r'_E \geq \phi_0(E, H) = r_E$. Similarly, we have $r'_X \geq r_X$. Combining Claim 19, we have

$$r_X - \mu r_{opt} \leq \underbrace{r'_X - \mu r_{opt} \leq r_E}_{\text{by Claim 19}} \leq \underbrace{r'_E \leq r_X + \mu r_{opt}}_{\text{by Claim 19}}. \tag{14}$$

So $|r_X - r_E| \leq \mu r_{opt}$, i.e., $\phi_0(E, H) \in \phi_0(X, H) \pm \mu r_{opt} \subseteq (1 \pm \mu)\phi_0(X, H)$. Therefore E is a μ-coreset of (X, d). ◀

4 Experiments

Our experimental results were obtained on a Windows workstation with 2.8GHz Intel(R) Core(TM) i5-840 and 8GB main memory; the algorithms were implemented in Matlab R2018a. We test our algorithms on both synthetic and real datasets. For Algorithm 2, we take two well known algorithms of k-center clustering with outliers, $Base1$ of [15] and $Base2$ of [33], as the baselines. For Algorithm 3, we compare our coreset construction with uniform random sampling.

To generate the synthetic datasets, we set $n = 10^5$ and $D = 10^3$, and vary the values of z and k. First, randomly generate k clusters inside a hypercube of side length 200, where each cluster is a random sample from a Gaussian distribution with variance 10; each cluster has a random number of points and we keep the total number of points to be $n - z$; we compute the minimum enclosing balls respectively for these k clusters (by using the algorithm of [6]), and randomly generate z outliers outside the balls. The maximum radius of the balls is used as r_{opt}.

We also use three real datasets. MNIST dataset [28] contains $n = 60,000$ handwritten digit images from 0 to 9, where each image is represented by a 784-dimensional vector. The 10 digits form $k = 10$ clusters. Caltech-256 dataset [29] contains $30,607$ colored images with 256 categories, where each image is represented by a 4096-dimensional vector. We choose $n = 2,232$ images of 20 categories to form $k = 20$ clusters. CIFER-10 training dataset [27] contains $n = 50,000$ colored images in 10 classes as $k = 10$ clusters, where each image is represented by a 4096-dimensional vector. For each real dataset, we use the minimum enclosing ball algorithm of [6] to compute r_{opt}, and randomly generate $z = 5\%n$ outliers outside the corresponding balls.

Results and analysis. Note that we exactly exclude z outliers (rather than $(1 + \epsilon)z$ as stated in Theorem 10 and 12) in our experiments, and calculate the approximation ratio $\phi_0(X, E)/r_{opt}$ for each instance, if E is the set of returned cluster centers.

We first run our Algorithm 1 on synthetic and real datasets. For synthetic datasets, we set $k =$ 2-20, and $\beta = |E|/k = 8$ via modifying the values of ϵ and η appropriately (that means we output $8k$ cluster centers); normally, we set $\eta = 0.1$ and $\epsilon \approx 0.7$. We try the instances with

$z = \{2\%n, 4\%n, 6\%n, 8\%n, 10\%n\}$, and report the average results in Figure 3 (a) and (b); the approximation ratios are within 1.3-1.4 and the running times are less than 30s. Actually, the performance is quite stable regarding different values of z in our experiments, and the standard variances of approximation ratios and running times are less than 0.03 and 0.12, respectively. We also vary the value of β from 4 to 28 with $k = 10$. Figure 3 (c) shows that the approximation ratio slightly decreases as β increases. The running times are all around 14s and do not reveal a clear increasing trend as β increases. We think the reason behind may be that we just use the simple $O(n \log n)$ sorting algorithm, rather than the linear time selection algorithm [10], for computing Q_j in practice (see Step 3(a) of Algorithm 1); thus the running time is not linearly dependent on $|E|$. The results for real datasets are shown in the full version of our paper; the approximation ratios are all below 1.3 and the running times are less than 35s even for the largest CIFER-10 dataset.

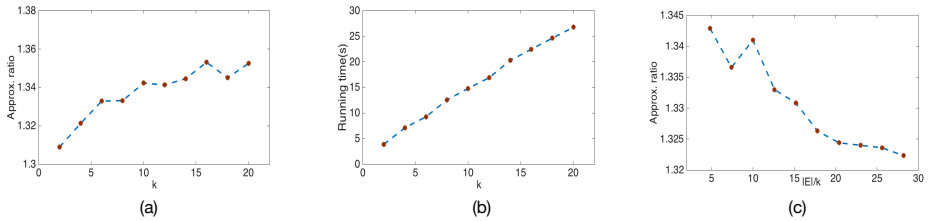

Figure 3 The experimental results of Algorithm 1 on synthetic datasets.

We also test our Algorithm 2 on synthetic and real datasets. We set $\epsilon = 1$ so that to avoid to repeat running Algorithm 2 too many times (see Corollary 13), but we still exactly exclude z outliers for calculating the approximation ratio as mentioned before. Our results are shown in Table 1. The synthetic and real datasets are too large for the baseline algorithms $Base1$ and $Base2$, e.g., they run too slowly or even out of memory in our workstation if n, z, and D are large (they have complexities $\Omega(n^2 D)$ or $\Omega(kznD)$)[3]. To make a fair comparison, we run $Base1$, $Base2$, and Algorithm 2 on smaller synthetic datasets with $(n = 2000, D = 10)$ and $(n = 2000, D = 100)$; we also set $z = \{2\%n, 4\%n, 6\%n, 8\%n, 10\%n\}$ as before and report the average results. When $D = 10$, $Base1$ and Algorithm 2 achieve approximation ratios < 1.5 generally (Figure 4 (a)); moreover, $Base2$ and Algorithm 2 run much faster than $Base1$ (Figure 4 (b)). However, when $D = 100$, $Base1$ and $Base2$ yield much worse approximation ratios than Algorithm 2 (Figure 4 (c) and (d)). Our experiment reveals that Algorithm 2 can achieve a more stable performance when dimensionality increases.

Finally, we compare the performances of our coresets method (Algorithm 3) and uniform random sampling in terms of reducing data sizes. Though real-world image datasets often are believed to have low intrinsic dimenions [8], it is difficult to compute them (e.g., doubling dimension) accurately. In practice, we can directly set an appropriate value for l in Step 1 of Algorithm 3 (without knowing the value of doubling dimension ρ). For example, the size of coreset is $2z + O\big((\frac{2}{\mu})^\rho k\big) = 2z + O(l)$ according to Theorem 18, so we keep the sizes of our coresets to be $\{15\%n, 20\%n, 25\%n\}$ via modifying the value of l in our experiments. Correspondingly, we also set the sizes of random samples to be $\{15\%n, 20\%n, 25\%n\}$. We run Algorithm 2 on the corresponding random samples and coresets, and report the results in Table 2. Running Algorithm 2 on the coresets yields approximation ratios close to

[3] We are aware of several distributed algorithms for k-center clustering with outliers [12, 19, 30, 32], but we only consider the setting with single machine in this paper.

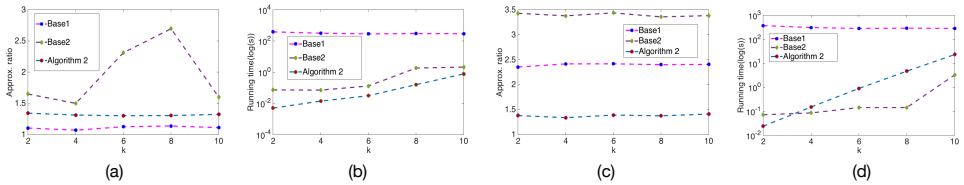

Figure 4 Comparison of Base1, Base2, and Algorithm 2 on smaller synthetic datasets ((a) and (b) for $D = 10$; (c) and (d) for $D = 100$).

Table 1 The results of Algorithm 2 on synthetic and real datasets.

	Synthetic datasets				Real datasets		
	k=2	k=4	k=6	k=8	MNIST	CALTECH256	CIFAR10
Approx. ratio	1.410	1.403	1.406	1.423	1.277	1.378	1.249
Running time(s)	8.097	63.636	374.057	1939.004	2644.709	2864.231	13295.306

those obtained by directly running the algorithm on the original datasets; the results also remain stably when the level reduces from 25% to 15%. More importantly, our coresets significantly reduce the running times (e.g., it only needs 15%-35% time by using 15%-level coreset). Comparing with the random samples, our coresets can achieve significantly lower approximation ratios especially for the 15% level. Note that the coreset based approach takes more time than uniform random sampling, because we count the time spent for coreset construction.

Table 2 The results of Algorithm 2 on random samples, coresets, and original datasets.

		random sampling			coreset			
		15%	20%	25%	15%	20%	25%	100%
MNIST	Appro. Ratio	1.591	1.597	1.566	1.275	1.261	1.261	1.277
	running time(s)	624.612	769.517	958.549	936.393	1071.926	1262.996	2644.709
CALTECH256	Appro. Ratio	2.144	1.779	1.448	1.431	1.420	1.401	1.378
	running time(s)	407.294	510.423	603.713	413.961	516.862	609.979	2864.231
CIFAR10	Appro. Ratio	1.538	1.383	1.446	1.248	1.256	1.249	1.249
	running time(s)	2420.943	2170.416	2938.773	3526.752	3264.858	4033.862	13295.306

5 Future Work

Following our work, several interesting problems deserve to be studied in future. For example, can the coreset construction time of Algorithm 3 be improved, like the fast net construction method proposed by Har-Peled and Mendel [20] in doubling metrics? It is also interesting to study other problems involving outliers by using greedy strategy.

References

1 Ankit Aggarwal, Amit Deshpande, and Ravi Kannan. Adaptive sampling for k-means clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 15–28. Springer, 2009.

2 Sepideh Aghamolaei and Mohammad Ghodsi. A Composable Coreset for k-Center in Doubling Metrics. In *Proceedings of the 30th Canadian Conference on Computational Geometry, CCCG 2018, August 8-10, 2018, University of Manitoba, Winnipeg, Manitoba, Canada*, pages 165–171, 2018.

3 Noga Alon, Seannie Dar, Michal Parnas, and Dana Ron. Testing of clustering. *SIAM Journal on Discrete Mathematics*, 16(3):393–417, 2003.

4 Noga Alon and Joel H Spencer. *The probabilistic method*. John Wiley & Sons, 2004.

5 Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*, 2017. `arXiv:1703.06476`.

6 Mihai Badoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2003.

7 Mihai Badoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 250–257, 2002.

8 Mikhail Belkin. *Problems of learning on manifolds*. The University of Chicago, 2003.

9 Avrim Blum, John Hopcroft, and Ravindran Kannan. Foundations of data science. *Vorabversion eines Lehrbuchs*, 2016.

10 Manuel Blum, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, 1973.

11 Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

12 Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. Solving k-center Clustering (with Outliers) in MapReduce and Streaming, almost as Accurately as Sequentially. *CoRR*, abs/1802.09205, 2018.

13 Deeparnab Chakrabarty, Prachi Goyal, and Ravishankar Krishnaswamy. The Non-Uniform k-Center Problem. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 67:1–67:15, 2016.

14 Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

15 Moses Charikar, Samir Khuller, David M Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 642–651. Society for Industrial and Applied Mathematics, 2001.

16 Moses Charikar, Liadan O'Callaghan, and Rina Panigrahy. Better streaming algorithms for clustering problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 30–39. ACM, 2003.

17 Amit Daniely, Nati Linial, and Michael E. Saks. Clustering is difficult only when it does not matter. *CoRR*, abs/1205.4891, 2012.

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

19 Sudipto Guha, Yi Li, and Qin Zhang. Distributed Partial Clustering. In *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 143–152. ACM, 2017.

20 Sariel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.

21 Sariel Har-Peled and Micha Sharir. Relative (p, ε)-approximations in geometry. *Discrete & Computational Geometry*, 45(3):462–496, 2011.

22 Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

23 Lingxiao Huang, Shaofeng Jiang, Jian Li, and Xuan Wu. Epsilon-Coresets for Clustering (with Outliers) in Doubling Metrics. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 814–825. IEEE, 2018.

24 Piotr Indyk, Sepideh Mahabadi, Mohammad Mahdian, and Vahab S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'14, Snowbird, UT, USA, June 22-27, 2014*, pages 100–108, 2014.

25 Anil K Jain. Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8):651–666, 2010.

26 Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):881–892, 2002.

27 Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

28 Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

29 Fei-Fei Li, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.

30 Shi Li and Xiangyu Guo. Distributed k-Clustering for Data with Heavy Noise. In *Advances in Neural Information Processing Systems*, pages 7849–7857, 2018.

31 Yi Li, Philip M Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *Journal of Computer and System Sciences*, 62(3):516–527, 2001.

32 Gustavo Malkomes, Matt J Kusner, Wenlin Chen, Kilian Q Weinberger, and Benjamin Moseley. Fast distributed k-center clustering with outliers on massive data. In *Advances in Neural Information Processing Systems*, pages 1063–1071, 2015.

33 Richard Matthew McCutchen and Samir Khuller. Streaming Algorithms for k-Center Clustering with Outliers and with Anonymity. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 165–178. Springer, 2008.

34 Jeff M. Phillips. Coresets and Sketches. *Computing Research Repository*, 2016.

35 Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 281–290, 2004.