New Challenges for Clustering: Massivenss and Uncertainty

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Master of Science

in

Computer Science and Technology

by

Xuan Wu

Thesis Supervisor: Associate Professor Jian Li

June 2018

New Challenges for Clustering: Massivenss and Uncertainty

by

Xuan Wu

Submitted to the Institute for interdisciplinary Information Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

at

TSINGHUA UNIVERSITY

June 2018

Author .....
Institute for interdisciplinary Information Sciences
April 18, 2018

Certified by .....
Jian Li
Associate Professor
Thesis Supervisor

Acknowledgments

First of all, I want to give special thanks to my advisor Dr.Jian Li. Three years ago, I was puzzled about my future and Dr.Li gave me the chance to stay in Tsinghua for graduate study. In the past three years, I have received extremely valuable experience in doing research with Dr.Li. Without Dr.Li, I can not even find my interest in theoretical computer science. I also want to thank his generous support in everything good to my research career. In particular, I appreciate his support in organizing the theoretical tea time and allowing me to visit Dr.Feldman in University of Haifa.

I want to thank my collaborators, Doctor Dan Feldman, Doctor Lingxiao Huang, Doctor Shaofeng H.-C Jiang, Shichuan Deng, Wenzheng Li and Changzhi Xie. Without them, most results in this paper can not appear.

Finally, I want to thank my parents. Their unconditional love and support give me the courage to pursue my dreams. Without the help of my father, I can not win the Olympiad of Mathematics and Informatics and get the chance to study in Yao Class, Tsinghua.

New Challenges for Clustering: Massivenss and Uncertainty

by

Xuan Wu

Submitted to the Tsinghua University
on April 18, 2018, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Clustering is a central problem in unsupervised learning and data analytics. Solving clustering problems over uncertain and massive data has become increasingly important in many applications and has attracted a lot of attention in recent years. In this paper, we focus on two new challenges of clustering: uncertainty and massiveness.

In many application scenarios, the precise locations of the points are not known before, but can be estimated through noisy measurements. To capture this, we propose a new model for clustering such uncertain data. In our model, there is a collection X of uncertain points in $\mathbb{R}^d$. Instead of knowing the precise location of every uncertain point $x \in X$, we can only estimate their locations through observations with noise. Upon each observation on an uncertain point x, we obtain an independent sample, which is assumed to follow a Gaussian distribution $N(x, I_d)$, where I_d is the $d \times d$ identity matrix. The objective is to compute clustering problem such as the optimal k-median clustering on X based on observations. We refer to the total number of observations as sample complexity which measures the statistical efficiency of an algorithm. we propose statistically and computationally efficient algorithms for computing approximate k-median clustering in this new model. In particular, we provide constant factor approximation algorithms with low sample complexity. To complement our algorithmic results, we also provide nearly matching sample complexity lower bounds.

We also consider the scenario where there is a large number of data points for the clustering task. In particular, the data may be too large to be read once. We study *robust coresets* for (k, z)-clustering with outliers. We show an improved connection between α-approximation and robust coreset. This also leads to improvement upon the previous best known bound of the size of robust coreset for Euclidean space [Feldman and Langberg, STOC 11, [39]]. The new bound entails a few new results in clustering and property testing.

Dissertation Supervisor: Associate Professor Jian Li

Contents

Chapter 1

Introduction

Clustering is a central problem in unsupervised learning, data analytics, and statistics [70, 80, 6, 32]. Among various objectives of clustering, the center based clustering is arguably the most popular one. Particularly, in this paper, we care about the (k, z)-clustering problems. In the (k, z)-clustering problem, the objective is to find a k-subset $C \in [X]^k$ (which are called centers), such that the objective function $\mathrm{dist}_z(X, C) := \sum_{x \in X} d^z(x, C)$ is minimized, where $d(x, C) := \min_{y \in C} d(x, y)$ is the distance from the point x to its closest center in C. In particular, $(k, 1)$-clustering is the well known k-median problem, $(k, 2)$-clustering the k-means problem, and (k, ∞)-clustering the k-center problem.

As the dramatically increasing of the data set volume, dealing with uncertain and massive data is becoming a novel challenging task. Focusing on these challenges, this paper consists of two main parts. To capture the task of clustering uncertain data, in Chapter 2, we propose a new model for clustering uncertain Euclidean points. To handle massive data, in Chapter 3, we study the notion of robust coreset for (k, z)-clustering with outliers. The robust coreset has applications in property testing, which is a representative task for handling massive data. We will discuss the two parts more precisely in following two sections.

1.1 New Model to Capture Clustering of Uncertain Data

Clustering uncertain data has attracted lots of attention from computer science community (see, e.g., [82], [59], [34], [52]). In this chapter, we propose a new model for clustering noisy data points in Euclidean space. In this model, the task is to cluster a set X of n uncertain data points in $\mathbb{R}^d$. Instead of having the precise coordinate of every point in X, we can only take observations on these points. We refer to the access to every point $x \in X$ as sample access $\mathcal{O}(x)$. In an observation of $\mathcal{O}(x)$, we take one sample from $N(x, I_d)$.

One important motivation of our model is the scenario where we have some observations on data points instead of having accurate representations of data points. In this case, we wonder how to cluster these data points efficiently while minimizing the cost of observations.

We discuss two practical scenarios.

Scenario One A scientific team needs to distribute a collection of battery-powered wireless sensors in Antarctica. Certain sensor networking protocol requires to dividing the sensors into k clusters. To obtain a good clustering, each sensor needs to report its coordinate to the central server. Each sensor has a positioning device (which has low battery power and inaccurate measures). Each position measurement returns a noisy sample, which is assumed to follow a Gaussian distribution centered at the sensor. Question: how to cluster the sensors efficiently meanwhile minimizing the total cost (i.e., the number of measurements)?

Scenario Two A recommendation system models every customer as a high dimensional point in Euclidean space and wants to divide the customers into some clusters. However, the system does not have the exact information of each customer. Instead, the system has many "realizations" of every customer. Here, a realization is a record of a customer's behavior, which reflects some partial information about the customer.

For example, her evaluation of a book or his recommendation of a book to a friend. Assume that there is a predictive machine learning algorithm that is used to estimate the customer's coordinate through his/her "realizations". Question: how to cluster the customers efficiently by using as few "realizations" as possible?

Among various clustering objectives, the center-based clustering is fundamentally important. The k-median clustering problem requires to partition the input data set X into k sets $X_1, ..., X_k$ and assign a center v_i to every X_i such that

$$\sum_{i=1}^{k} \sum_{x \in X_i} \|x - v_i\| \tag{1.1}$$

is minimized over all possible partitions and centers, where $\|\cdot\|$ is the Euclidean norm. We consider the following noisy version of k-median in this chapter.

Definition 1.1.1 (Noisy k-median Problem)**.** Let $\mathcal{O}(x)$ denote the sample access to $N(x, I_d)$, i.e., the standard Gaussian distribution centered at x. Given sample access $\{\mathcal{O}(x_j) : j = 1, 2, ..., n\}$ to n uncertain points. In every step, we are allowed to choose one $\mathcal{O}(x_i)$ and take a sample from it. The goal is to obtain a constant factor approximation of the optimal k-median value

$$\text{OPT} = \min_{V:|V|=k} \sum_{i=1}^{N} \min_{v \in V} \|x_j - v\|.$$

and the corresponding approximate clustering centers, with probability at least $1 - \delta$, while taking as few samples as possible.

Beyond (approximately) finding the optimal k-median value, we additionally require to find the corresponding centers. We want to remark that one can deduce from our algorithms and proofs that the additional task is a by-product. However, in many practical scenarios, finding the centers can be beneficial. In particular, the centers are the compression of the data so that one can quickly answer the question "which cluster does a point belong to?" with the help of centers.

Our work is closely related to the stochastic multi-armed bandit setting, which is a classical model for characterizing the exploration-exploitation trade-off when the

environment is stochastic. In the usual setting of multi-armed bandit, we are given n stochastic arms, each associated with an unknown distribution of reward. In each step, we can pick an arm and get a reward sampled from its corresponding distribution. The typical objective for a multi-armed bandit instance includes maximizing the cumulative sum of rewards or minimizing the cumulative regret (see e.g., [14, 17]). Our work bears more resemblance with the so-called *pure exploration* setting, where the objective is to identify the optimal solution (or an approximate optimal one) with high-confidence while using as few samples as possible. This setting has attracted significant attention by its wide applications in medical trials, crowdsourcing, communication network, databases and online advertising [16, 29, 83]. Due to the relationship to the multi-armed bandit model, we also call the sample access $\mathcal{O}(x)$ ($x \in \mathbb{R}^d$) an arm for simplicity.

Unlike in the usual setting of multi-armed bandit, one arm in our setting corresponds to a multi-dimensional distribution (in $\mathbb{R}^d$) instead of a reward distribution (in $\mathbb{R}$). If $d = 1$, our model is actually the multi-armed bandit model. To the best of our knowledge, there is no existing result about clustering in the multi-armed bandit model. In this chapter, we will discuss the 1-median clustering in the multi-armed bandit model as a special case (Section 2.2) and provide an algorithmic result (Theorem 2.2.6) and a lower bound result (Theorem 2.4.1) which together gives the almost optimal sample complexity.

1.1.1 Main Contribution

We provide algorithmic results and lower bound results for Noisy k-median (Problem 1.1.1).

Theorem 1.1.2 (Informal statements of Theorems 2.1.3, 2.3.4 and 2.4.5)**.** *Let* $X = \{x_1, ..., x_n\} \subset \mathbb{R}^d$. *Let* $OPT = \min_{V:|V|=k} \sum_{i=1}^{N} \min_{v \in V} \|x_j - v\|$. *There is an algorithm* noisyKmedian*, given sample access* $\mathcal{O}(x_j)$ *(*$j \in [n]$*), outputs an* $O(1)$*-approximate* k*-median clustering and an* $O(1)$*-approximate* k*-median value on* X*, with sample*

complexity

$$\tilde{O}(d(n^3\mathrm{OPT}^{-2}+n)).$$

There is another algorithm noisyKmedian2*, given sample access to* $\mathcal{O}(x_j), j \in [n]$*, outputs an* $O(1)$*-approximate* k*-median clustering and an* $O(1)$*-approximate* k*-median value on* X*, with sample complexity*

$$\tilde{O}(dk^2(n^2\mathrm{OPT}^{-2}+n)).$$

On the other hand, every algorithm which computes a 2*-approximation for every noisy* k*-median instance requires at least*

$$\Omega(n^2\mathrm{OPT}^{-2}+n)$$

many samples.

Another natural question for our model also arises: what is the best approximation ratio, e.g., can it be $1+\varepsilon$ for any $\varepsilon > 0$? What is the sample complexity for achieving a better approximation ratio? This question has a simple positive answer: by noisyKmedian or noisyKmedian2, we can obtain an $O(1)$-approximation for OPT. We then estimate each uncertain point up to an error $\varepsilon\mathrm{OPT}/n$ and run any existing k-median algorithms on these estimations, e.g., exhaustive search [1]. The sample complexity is $O(d\varepsilon^{-2}n^3\mathrm{OPT}^{-2})$. In Appendix B, we propose another algorithm with the sample complexity

$$\tilde{O}\left(d(n^3\mathrm{OPT}^{-2}+\varepsilon^{-4}kn^2\mathrm{OPT}^{-2}+n)\right)$$

which is more efficient when $n \gg k\varepsilon^{-2}$. The main idea is to first construct a coreset and then run existing k-median algorithms on the coreset.

[1] If either k or d is not constant, exhaustive search requires exponential time. If one cares about the running time, we can use other known polynomial time approximation algorithms, like BPRS ([15]).

1.1.2 Technique Overview

Overview of Our NoisyKMedian Algorithm Our algorithm NoisyKMedian is UCB-based (Upper confidence Bound), i.e., it maintains a confidence ball on each uncertain point. Precisely, at the i-th round, we estimate every uncertain point $x \in X$ to $O(2^{-i})$ in the Euclidean distance and compute the approximate optimal k centers on these estimates. Then we check if current centers are actually an $O(1)$-approximate k-median clustering. Our checking process involves the problem of computing the distance from n uncertain points to k centers, which has a delicate structure. If an uncertain point is very close to a center, we need to spend many samples to estimate the distance. To overcome this problem, we design a UCB-based algorithm DisNATKP to omit those distances which are too small compared to the total distance.

Overview of Our NoisyKMedian2 Algorithm Our algorithm NoisyKMedian2 is based on an inherently different idea. This algorithm is testing-based and designed to achieve a better sample complexity on the parameter n. To achieve this goal, we need a process with the same function as DisNATKP but with tighter sample complexity on n. In fact, we construct such a process called TestNATKP, based on a sampling technique which estimates the contribution of those data points close to the given k centers and takes much fewer samples than DisNATKP. Then based on TestNAPKP, we show that given a noisy k-median instance and a number $C > 0$, there is an algorithm TestKmedian to test whether the optimal k-median value is larger than $10C$ or smaller than $C/10$. At a high level, our testing algorithm guesses arms which can be an estimation of the optimal centers and estimates the locations of those arms by taking enough samples. However, our testing algorithm needs to call TestNAPKP at least $\Omega(n^k)$ times. Hence Algorithm TestKmedian requires exponential time in k which is a weakness compared to NoisyKMedian. Finally, we construct a simple binary search procedure NoisyKMedian2 by calling TestKmedian to decide the value of (approximate) optimal solution.

Overview of Our Lower Bound We provide both the instance lower bound and worst-case lower bound for the noisy k-median problem. For our instance lower bound $\Omega(n^2\mathrm{OPT}^{-2} + n)$, we use the "Change of Distribution" lemma (Lemma 1.4.3) and follow a classical framework for lower bounding the sample complexity of randomized algorithms. We perturb the data points and obtain a new instance such that the optimal k-median value changes significantly. Then we use the "Change of Distribution" lemma to lower bound the sample complexity of any algorithm which distinguishes the two instances. Especially, when $d = k = 1$, our instance lower bound (Theorem 2.4.5) matches the algorithmic upper bound (Theorem 2.1.3).

For our worst-case lower bound, we construct a sequence of noisy k-median instances which require $\Omega(\sqrt{d}n^2\mathrm{OPT}^{-2}+n)$ samples. It implies that the factor d is also important and cannot be omitted in the sample upper bound. Our main approach is to reduce the problem to the statistical task of distinguishing the case that the center of a normal distribution is 0 or significantly larger than 0 (see Lemma 2.4.1).

1.2 Handling Massive Data: Robust Coreset and Property Testing

A powerful technique for solving the (k, z)-clustering problem is to construct coresets [56, 28, 39, 42]. A coreset is a weighted subset of the point set, such that for any set of k centers, the objective function computed from the coreset is approximately the same as that computed from all points in X. Hence, a coreset can be used as proxy for the full data set: one can apply the same algorithm on the coreset, and the result on the coreset approximates that on the full data set.

Definition 1.2.1. An ε-coreset for the (k, z)-clustering problem in metric space $M(X, d)$ is a weighted subset S of X with weight $w : S \to \mathbb{R}_{\geq 0}$ [2], such that for

[2]Some work may allow the weight to be negative, but we require it to be nonnegative in our work.

any k-subset $C \in [X]^k$,

$$\sum_{x \in S} w(x) \cdot d^z(x, C) \in (1 \pm \varepsilon) \cdot \mathcal{K}_z(X, C).$$

Typically, we require that the size of the coreset depends on $1/\varepsilon$, k and z (independent of $|X|$). Apparently, a small coreset is much cheaper to store and can be used to estimate the objective function more efficiently. In fact, constructing coresets can be useful in designing more efficient approximation algorithms for many clustering problems, with various constraints and outliers [39, 43, 42, 13, 45, 71].

However, constructing coresets needs at least reading the whole data set once. When dealing with massive data, especially when the data becomes to large to be read once, we require the notion of robust coreset which is a relaxed version of coreset. The most important advantage of robust coreset is that it can be constructed by uniform sampling (and hence in sublinear time)The notion of robust coreset was first introduced in [39]. In the following, we give the definition of robust coreset for the (k, z)-clustering problem with outliers.

Definition 1.2.2 (robust coresets)**.** Let $M(X, d)$ be a metric space. Let $0 < \gamma \leq 1$, $0 \leq \varepsilon, \alpha \leq \frac{1}{4}$, $k \geq 1$ and $z > 0$. For any $H \subseteq X$ and $C \in [X]^k$, let

$$\mathcal{K}_z^{-\gamma}(H, C) := \min_{H' \subseteq H : |H'| = \lceil (1-\gamma)|H| \rceil} \sum_{x \in H'} d^z(x, C)$$

denote the sum of the smallest $\lceil (1-\gamma)|H| \rceil$ values $d^z(x, C)$ over $x \in H$ (i.e., we exclude the largest $\gamma|H|$ values as outliers). An (α, ε)-robust coreset for the (k, z)-clustering problem with outliers is a subset $S \subseteq X$ such that for any k-subset $C \in [X]^k$ and any $\alpha < \gamma < 1 - \alpha$,

$$(1 - \varepsilon) \cdot \frac{\mathcal{K}_z^{-(\gamma+\alpha)}(X, C)}{|X|} \leq \frac{\mathcal{K}_z^{-\gamma}(S, C)}{|S|} \leq (1 + \varepsilon) \cdot \frac{\mathcal{K}_z^{-(\gamma-\alpha)}(X, C)}{|X|}.$$

Our result for robust coreset for (k, z)-clustering is presented in the following theorem, which generalizes and improves the prior result in [39] for Euclidean space.

Beyond Euclidean Space, we also generalize the robust coreset to metric space with bounded doubling dimension which is arguably the most popular notion to capture the complexity of metric space. Precisely, a metric space $M(X, d)$ has doubling dimension t, if t is the smallest number such that every ball in X can be covered by at most 2^t balls of half the radius [8, 53]. We denote the doubling dimension by $\mathrm{ddim}(M)$. The doubling dimension measures the intrinsic dimensionality of a general metric space, and it generalizes the dimension of normed vector spaces, where t-dimensional ℓ_p space has doubling dimension $O(t)$ [8]. Many problems have been studied in doubling metrics, such as spanners [46, 33, 50, 51, 19, 25, 24, 77, 20], metric embedding [53, 1, 21], nearest neighbor search [31, 60, 57], and approximation algorithms [79, 10, 18, 23, 22, 45].

Theorem 1.2.3 (informal, robust coreset)**.** *Let $M(X, d)$ be a doubling metrics (a d-dimensional Euclidean space resp.). Let S be a uniform sample of size $\tilde{O}(k \cdot \mathrm{ddim}(M)/\alpha^2)$ ($\tilde{O}(kd/\alpha^2)$ resp.) from X. Then with constant probability, S is an (α, ε)-robust coreset ($(\alpha, 0)$-robust coreset resp.) for the (k, z)-clustering problem with outliers.*

The definition of robust coreset in [39] is slightly different from ours. [3] One can directly check that in Euclidean space, an $(\gamma\varepsilon/4, 0)$-robust coreset in Definition 1.2.2 is an (γ, ε)-coreset in [39, Definition 8.1]. Thus the above theorem improves the size of (γ, ε)-coreset in [39, Corollary 8.4] from $\tilde{O}(kd\gamma^{-2}\varepsilon^{-4})$ to $\tilde{O}(kd\gamma^{-2}\varepsilon^{-2})$.

Furthermore, we demonstrate an application of robust coresets in property testing (Section 3.2). We design a simple testing algorithm for (k, z)-clustering. Alon et al. [4] first considered the property testing problem in the context of clustering. In particular, they studied the testing algorithm for k-center clustering. In this paper, we use robust coreset to develop a unified testing algorithm for (k, z)-clustering (for constant k and z). The testing algorithms can be converted into a sublinear time approximation algorithms for clustering with outliers. As pointed out in [4], one

[3] In [39, Definition 8.1], $S \subset X$ is called a (γ, ε)-coreset if for every $C \in [X]^k$, $\gamma_1 \geq \gamma$ and $\varepsilon_1 \geq \varepsilon$, $(1-\varepsilon_1) \cdot \frac{1}{|X|}\mathcal{K}_1^{-(1-\gamma_1+\varepsilon_1\gamma_1)}(X, C) \leq \frac{1}{|S|}\mathcal{K}_1^{-(1-\gamma_1)}(S, C) \leq (1+\varepsilon_1) \cdot \frac{1}{|X|}\mathcal{K}_1^{-(1-\gamma_1-\varepsilon_1\gamma_1)}(X, C)$.

interesting benefit of such algorithms is that they can answer the query "which cluster does a data point belong to", without really partition all the data points.

Constructing robust coresets is also a useful subroutine in several other problems, such as robust median and bi-criteria approximation for projective clustering (see [39]). Hence, our improvement may lead to certain improvements of these problems as well. Since this is not the focus of the this paper, we do not go into the details.

1.3 Related Work

Deterministic k-median clustering has attracted a lot of attention. Charikar et al. [26] gave the first constant factor ($\frac{20}{3}$) approximation algorithm by LP-rounding. Jain and Vazirani [62] improved the constant factor to 6 by reducing the k-median problem to the Uncapacitated Facility Location (UFL) problem. The approximation ratio was further improved to $3+\varepsilon$ by the well-known local search heuristics [61, 7]. The current best approximation ratio achieved by [15] is $2.675+\varepsilon$, based on a breakthrough work [68].

Clustering problems in different stochastic settings have been studied before, such as the locational uncertainty model and the existential uncertainty model. In both stochastic geometry models, the distribution of each data point is known previously. Many clustering problems have been studied in such models. Feldman and Langberg [40] considered the k-median problem, the j-flat-median (i.e., subspace approximation) problem, and the k-line-median problem in the deterministic setting. Note that their techniques also work for the stochastic variants. Huang and Li [59] investigated the stochastic k-center and j-flat-center problems in both stochastic geometry models.

Recently, Mazumdar and Saha [72] also considered clustering with noisy queries. However, their model is inherently different from ours. The oracle in their model answers query of the the form "Do points i and j belong to the same cluster" and returns the correct answer with certain probability. Moreover, they study the problem of exactly reconstructing the underling clustering instead of computing an approximated clustering.

Feldman and Langberg [39] first studied the notion of robust coreset to handle the clustering problems with outliers. In $\mathbb{R}^d$, they showed how to construct a (γ, ε)-coreset [4] of size $\tilde{O}(kd\varepsilon^{-4}\gamma^{-2})$ by uniform sampling. We improve the bound to $\tilde{O}(kd\varepsilon^{-2}\gamma^{-2})$. Later, Feldman et al. [43] developed another notion called *weighted coreset* to handle outliers. They used such coresets to design an $(1+\varepsilon)$-approximation algorithm for the k-median problem with outliers.

In the seminal paper [2], Agarwal et al. proposed the notion of coresets for the directional width problem (in which a coreset is called an ϵ-kernel) and several other geometric shape-fitting problems. Since then, coresets have become increasingly more relevant in the era of big data as they can reduce the size of a dataset with provable guarantee that the answer on the coreset is a close approximation of the one on the whole dataset. Many efficient algorithms for constructing small coresets for clustering problems in Euclidean spaces are known (see e.g., [3, 54, 28, 55, 67, 39, 42, 13]). In particular, Feldman and Langberg [39] showed a construction for ε-coresets of size $\tilde{O}(dk/\varepsilon^{2z})$ for general (k, z)-clustering problems with arbitrary k and z, in $\tilde{O}(nk)$ time. For the special case that $z = 2$ which is the k-means clustering, Braverman et al. [13] improved the size to $\tilde{O}(k^2 \min\{k/\varepsilon, d\}/\varepsilon^2)$, which is *independent* of the dimensionality d. For another special case $z = \infty$, which is the k-center clustering, an ε-coreset of size $O(k/\varepsilon^d)$ can be constructed in $O(n + k/\varepsilon^d)$ time, for $\mathbb{R}^d$ [3, 54]. For another special case $z = \infty$, which is the k-center clustering, an ε-coreset of size $O(k/\varepsilon^d)$ can be constructed in $O(n+k/\varepsilon^d)$ time, for $\mathbb{R}^d$ [3, 54]. For general metrics, an ε-coreset for the (k, z)-clustering problem of size $O(k \log n/\varepsilon^{2z})$ can be constructed in time $\tilde{O}(nk)$ [39]. We refer interested readers to Phillips's survey [73] for more construction algorithms as well as the applications of coresets in many other areas.

Property Testing is proposed in the seminal work of [75] and [47], which is generally the study of designing and analyzing of randomized decision algorithm on efficiently making decision whether the given instance is having certain property or

[4] Note that their definition [39, Definition 8.1] is similar but slightly different to ours. However, considering the (k, z)-clustering problem with outliers, one can check that an $(\varepsilon\gamma/4, \varepsilon)$-robust coreset in our Definition 1.2.2 is a (γ, ε)-coreset in [39, Definition 8.1]. In fact, our defintion is more general. It is unclear whether their result applies to our definition.

somewhat far from having it. Significantly, the query complexity of efficient property testing algorithm is often sublinear on the size of its accessing instance. Many important properties have been studied in the context of property testing, such as linearity ([12],[11],[49],[76],[35]), low-degree([4], [5],[63],[64],[74]), and monotonicity ([37],[38],[44],[48],[30]).

1.4 Preliminaries

Recall that $\mathcal{O}(x)$ ($x \in \mathbb{R}^d$) denotes the sample access to uncertain point x with Gaussian noise, i.e., a sample of $\mathcal{O}(x)$ follows from the distribution $N(x, I_d)$. Since our model is a high dimensional variant of the multi-armed bandit model, we also call $\mathcal{O}(x)$ an arm for simplicity. The distance from an arm $\mathcal{O}(x)$ to a point $p \in \mathbb{R}^d$ is defined to be $\|x - v\|$ where $\|\cdot\|$ is the Euclidean norm.

We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. For a point $x \in \mathbb{R}^d$ and a set of points V, the distance between x and V is defined to be $\mathrm{dist}(x, V) = \min_{v \in V} \|x - v\|$. For a set of points $X \subset \mathbb{R}^d$, we define $\mathrm{cost}(X, V) = \sum_{x \in X} \mathrm{dist}(x, V)$. When $V = \{v\}$, with a little abuse of notation we also refer to $\mathrm{cost}(X, V)$ as $\mathrm{cost}(X, v)$. Also in what follows, $B(x, r)$ for $x \in \mathbb{R}^d, r > 0$ denotes an open ball centered at x with radius r. We say a point x or an arm $\mathcal{O}(x)$ is C-far from a point p if $\|x - p\| \geq C$.

An algorithm is called an α-approximation algorithm for k-median on X if it computes a set of k centers such that $\mathrm{cost}(X, V) \leq \alpha \mathrm{OPT}$ where OPT is the optimal k-median value on X. A randomized algorithm is called δ-correct if it succeeds with probability at least $1 - \delta$.

We use $\tilde{O}_\alpha(f)$ to hide polylogarithmic factor on α and f. Precisely, $\tilde{O}_\alpha(f(x))$ denotes a variable in $O\Big(f(x)\mathrm{poly}\big(\log f(x), \log \alpha\big)\Big)$. We need the following classical concentration inequality for Gaussian vectors.

Theorem 1.4.1 (See e.g. [81])**.** *Let x be a standard Gaussian vector centered at $\mu \in \mathbb{R}^d$, i.e., x is taken from $N(\mu, I_d)$. Then*

$$\Pr(\big|\|x - \mu\| - \sqrt{d}\big| \geq \varepsilon) \leq 2\exp\big(-c\varepsilon^2\big)$$

for some constant $c > 0$.

We also need the following classical additive Chernoff bound.

Theorem 1.4.2 (Chernoff Bound)**.** *Let $\varepsilon > 0$ be some constant. Let $x_1, x_2, \ldots, x_m$ be i.i.d 0-1 random variables and $\mu = \mathbb{E}x_1$. Then*

$$\Pr\left(\left|\frac{1}{m}\sum_{i=1}^{m} x_i - \mu\right| \geq \varepsilon\right) \leq 2e^{-2m\varepsilon^2}.$$

Change of Distribution. The following "Change of Distribution" lemma, formulated by [65], characterizes the behavior of an algorithm when underlying distributions of the arms are slightly altered, and is thus useful for proving sample complexity lower bounds. In the following, $\Pr_{\mathbb{A},\mathcal{C}}$ and $\mathbb{E}_{\mathbb{A},\mathcal{C}}$ denote the probability and expectation when algorithm $\mathbb{A}$ runs on instance $\mathcal{C}$.

Lemma 1.4.3 (Change of Distribution)**.** *Let $\mathbb{A}$ be an algorithm that runs on n arms, and let $\mathcal{C} = (a_1, a_2, \ldots, a_n)$ and $\mathcal{C}' = (a'_1, a'_2, \ldots, a'_n)$ be two sequences of n arms. Let random variable τ_i denote the number of samples taken from the i-th arm. For any event $\mathcal{E}$ in $\mathcal{F}_\tau$, where τ is a stopping time with respect to the filtration $\{\mathcal{F}_t\}_{t\geq 0}$, it holds that*

$$\sum_{i=1}^{n} \mathbb{E}_{\mathbb{A},\mathcal{C}}[\tau_i]\,\mathrm{KL}\left(a_i, a'_i\right) \geq d\left(\Pr_{\mathbb{A},\mathcal{C}}[\mathcal{E}], \Pr_{\mathbb{A},\mathcal{C}'}[\mathcal{E}]\right).$$

Let $\mathrm{KL}(a_1, a_2)$ denote the Kullback-Leibler divergence from the distribution of arm $a_2 = \mathcal{O}(x_2)$ to that of arm $a_1 = \mathcal{O}(x_1)$. We will need the following fact when using the above lemma in this paper.

$$\mathrm{KL}(N(x_1, I_d), N(x_2, I_d)) = \frac{1}{2}\|x_1 - x_2\|^2. \tag{1.2}$$

Deterministic k-Median Algorithm. In this paper, we use as subroutine an algorithm in [15], which has the current best approximation guarantee for k-median. We formalize the result here.

Theorem 1.4.4. *There is a polynomial time algorithm* $\mathsf{BPRS}(\cdot)$ *which receives a set* X *of deterministic points in* $\mathbb{R}^d$ *and outputs an* α*-approximation* k*-median clustering for* $\alpha < 2.676$.

Definition 1.4.5 (doubling dimension)**.** A metric space has doubling dimension at most t, if any ball can be covered by at most 2^t balls of half the radius. The doubling dimension of a metric space M is denoted as $\mathrm{ddim}(M)$.

We adapt the function representation used in [39, Definition 7.2], but specifically tailored to our own needs. In particular, since we focus on the clustering problems in a doubling metric $M(X, d)$, the ground set is $[X]^k$ (the set of k-subsets) throughout the paper. When $k = 1$, we use X to represent $[X]^1$ for simplicity.

We mainly focus on range spaces induced by a metric space. Hence we always consider *indexed* function sets. A set of functions $\mathcal{F}$ is called *indexed*, if there exists an index set V such that $\mathcal{F} = \{f_x \mid x \in V\}$. In most cases, we simply use $V = X$ as the index set.

Range Space. Let $\mathcal{F}$ be an indexed function set. Define $\mathrm{range}(\mathcal{F}, C, r) := \{f_x \in \mathcal{F} \mid f_x(C) \leq r\}$ for $C \in [X]^k, r \geq 0$. Define $\mathrm{ranges}(\mathcal{F}) := \big\{\mathrm{range}(\mathcal{F}, C, r) \mid C \in [X]^k, r \geq 0\big\}$ to be the collection of all the range sets. The range space of $\mathcal{F}$ is defined as the pair $(\mathcal{F}, \mathrm{ranges}(\mathcal{F}))$.

Now, We define the dimension of a range space, following [39].

Definition 1.4.6 ((shattering) dimension of a range space)**.** Suppose $\mathcal{F}$ is an indexed function set with ground set $[X]^k$. The (shattering) dimension of the range space $(\mathcal{F}, \mathrm{ranges}(\mathcal{F}))$, or simply the (shattering) dimension of $\mathcal{F}$, denoted as $\dim(\mathcal{F})$, is the smallest integer t, such that for any $\mathcal{D} \subseteq \mathcal{F}$ with $|\mathcal{D}| \geq 2$, $|\mathrm{ranges}(\mathcal{D})| \leq |\mathcal{D}|^t$. We note that in $\mathrm{ranges}(\mathcal{D})$, the same ground set $[X]^k$ is implicit.

We need a well studied notion in the PAC learning theory, called α-approximation.

Definition 1.4.7 (α-approximation of a range space)**.** Given a range space $(\mathcal{F}, \mathrm{ranges}(\mathcal{F}))$ (with ground set $[X]^k$), a set $\mathcal{S} \subseteq \mathcal{F}$ is an α-approximation of the range space, if for

every $\text{ranges}(\mathcal{F}, C, r) \in \text{ranges}(\mathcal{F})$ $(C \in [X]^k, r \geq 0)$

$$\left| \frac{|\text{range}(\mathcal{F}, C, r)|}{|\mathcal{F}|} - \frac{|\mathcal{S} \cap \text{range}(\mathcal{F}, C, r)|}{|\mathcal{S}|} \right| \leq \alpha.$$

In particular, it was shown that a small sized (depending on α and the VC dimension[5]) independent sample from the function set is an α-approximation with constant probability (see for example [69]).

[5]Our definition of the dimension is the shattering dimension of a range space, which tightly relates to the VC-dimension (see for example [66]). In particular, if $\dim(\mathcal{F})$ is t, then the VC-dimension of $\mathcal{F}$ is bounded by $O(t \log t)$.

Chapter 2

The Sample Complexity of Stochastic k-median Problem

This chapter is devoted to the noisy k-median problem 1.1.1. In Section 2.1, we present and analyze the first approximation algorithm NoisyKMedian. For the simplest case $k = d = 1$, we present an algorithm with lower sample complexity in Section 2.2. Then we apply our technique in Section 2.2 to the general setting and obtain the second algorithm NoisyKMedian2 in Section 2.3. In Section 2.4, we study both the instance-wise and worst-case lower bound for noisy k-median. Appendix A includes all missing proofs in the main text, and Appendix B includes our coreset construction algorithm.

2.1 An Upper Confidence Bound(UCB)-based Algorithm

In this section, we first develop an algorithm to compute the sum of distances from n arms to fixed k centers. Then, we show how to solve the noisy k-median problem based on this algorithm.

2.1.1 Computing Sum of Distances from n Arms to k Points

We introduce an UCB-based algorithm for computing the sum of distances from n arms to fixed k centers. This algorithm is a critical process in Algorithm 2 which solves the noisy k-median problem.

Intuitively, we can compute distances between every arm-center pair in parallel. The difficulty is that if the distance from an arm to a center is too small, we need a huge number of samples to compute a constant factor approximation for this distance. To deal with this problem, our algorithm DisNATKP iteratively estimates every distance from an arm to the fixed set V of k centers. In the process of the algorithm, we omit all remaining small distances once we confirm that their contributions to the total sum can be safely neglected.

Before stating the result, we highlight the critical steps in DisNATKP. Our algorithm iteratively estimates the distance from every arm to V and always maintains an estimation D of the total sum. At the beginning of each iteration, the algorithm checks if the contribution of all remaining arms can be safely neglected compared to D in Line 3. If it is not the case, the algorithm starts a new iteration. In Line 4, the algorithm decides the number of samples for each arm in i-th round, say m_i. In Line 5, the algorithm estimates x_j for each arm by the average $x_j^{(i)}$ of m_i samples taken from $\mathcal{O}(x_j)$. Observe that m_i increases exponentially in each round by Lines 4 and 12. Hence the estimation $x_j^{(i)}$ is guaranteed to be more and more accurate. From Line 6 to Line 11, the algorithm checks for each j whether $x_j^{(i)}$ is already a good approximation for the distance between x_j and V. If it is the case, the algorithm adds $\mathrm{dist}(x_j^{(i)}, V)$ to the total sum of distances D.

Lemma 2.1.1. *Let* $\mathrm{DIS} = \sum_{j\in[n]} \mathrm{dist}(x_j, V)$. *The algorithm* ***disNATKP***(X, V, δ) *takes at most*

$$\tilde{O}_{\delta^{-1}}\left(d(n^3\mathrm{DIS}^{-2} + n)\right)$$

many samples and outputs $D \in [\frac{2}{5}\mathrm{DIS}, \frac{3}{2}\mathrm{DIS}]$, *with probability at least* $1-\delta$.

Proof. Define events $\mathcal{E}_i = \{\omega : \forall j \in [n], \|x_j^{(i)} - x_j\| \le r_i\}$, i=1,2,... By the definition of $x_j^{(i)}$, we know that $\sqrt{m}x_j^{(i)}$ follows a Gaussian distribution $N(\sqrt{m}x_j, I_d)$. Then by

Algorithm 1 DisNATKP($X = \{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, V, \delta$)

Require: A sample access $\mathcal{O}(x_i)$ to $N(x_i, I_d)$ for each $i \in [n]$, a set V of k centers and a confidence parameter $\delta \in (0, 1)$.

Ensure: A number D as an approximation of $\sum_{i=1}^{n} \mathrm{dist}(x_i, V)$.

$r_1 \leftarrow 1$, $i \leftarrow 1$, $D \leftarrow 0$, and $T \leftarrow n$.
For every $j \in [n]$, flag(j) $\leftarrow$FALSE.
while $48Tr_i \geq D$ **do**
  $m_i = O\big(r_i^{-2}(d + \log(\delta^{-1}ni))\big)$.
  Take m_i samples from every arm $\mathcal{O}(x_j)$ $(j \in [n])$ and compute the average $x_j^{(i)}$ as an estimate of x_j.
  for $j = 1, 2, \ldots, n$, flag(j) =FALSE **do**
    For every $v \in V$, compute $c_{jv} = \max_{y \in B(x_j^{(i)}, 3r_i)} \|y - v\|$ and $d_{jv} = \min_{y \in B(x_j^{(i)}, 3r_i)} \|y - v\|$.
    if $\big(\forall v \in V, d_{jv} > 0$ and $\exists v_1 \forall v_2 \neq v_1, c_{jv_1} \leq 2d_{jv_2}\big)$ **then**
      flag(j) $\leftarrow$TRUE, $D \leftarrow D + \mathrm{dist}(x_j^{(i)}, V)$, $T \leftarrow T - 1$.
    end if
  end for
  $r_{i+1} \leftarrow r_i/2$, $i \leftarrow i + 1$.
end while
return D.

Theorem 1.4.1, we have

$$\Pr\left[\|\sqrt{m}(x_j^{(i)} - x_j)\| \geq \sqrt{m}r_i\right] \leq \Pr\left[\left|\|\sqrt{m}(x_j^{(i)} - x_j)\| - \sqrt{d}\right| \geq \sqrt{m}r_i - \sqrt{d}\right] \leq \frac{\delta}{3ni^2}. \tag{2.1}$$

Then by the union bound, we have

$$\Pr[\mathcal{E}_i] \geq 1 - \sum_{j \in [m]} \Pr\left[\|x_j^{(i)} - x_j\| \geq r_i\right] \overset{\text{Eq. (2.1)}}{\geq} 1 - n \cdot \frac{\delta}{3ni^2} = 1 - \frac{\delta}{3i^2}.$$

Let $\mathcal{E} = \cap_{i \geq 1} \mathcal{E}_i$. We have

$$\Pr[\mathcal{E}] \geq 1 - \frac{\delta}{3}(1 + \frac{1}{2^2} + \frac{1}{3^2} + \ldots) \geq 1 - \delta$$

Next, we need the following lemma for preparation. It's a very fundamental argument for UCB-based algorithm, and the proof can be found in Appendix A.

Lemma 2.1.2. *Conditioned on $\mathcal{E}$, for any integer $i > 0$ and $j \in [n]$, if $r_i \leq \mathrm{dist}(x_j, V)/12$, then $\mathrm{flag}(j)$ has been set to be "TRUE" at the i-th round. If $r_i > \mathrm{dist}(x_j, V)/2$, then $\mathrm{flag}(j)$ remains "FALSE" at the i-th round. Consequently, if $\mathrm{flag}(j)$ is set to be "TRUE" at the i-th round, then $\mathrm{dist}(x_j^{(i)}, V) \in [\frac{1}{2}\mathrm{dist}(x, V), \frac{3}{2}\mathrm{dist}(x_j, V)]$.*

Now we come back to prove Lemma 2.1.1. Conditioned on $\mathcal{E}$, denote by TR the set of $j \in [n]$ such that $\mathrm{flag}(j)$ is "TRUE" when the algorithm terminates. Let $\mathrm{FL} = [n] \setminus \mathrm{TR}$ be its complement. For $j \in \mathrm{TR}$, assume $\mathrm{flag}(j)$ is set to be "TRUE" at the i_j-th round of the while-loop. In what follows, we assume variables i, T, D and r_i are taking their values when the algorithm terminates (which means the condition of Line 3 is not satisfied). Then we have,

$$D = \sum_{j \in \mathrm{TR}} \mathrm{dist}(x_j^{(i_j)}, V) \overset{\text{Lemma 2.1.2}}{\leq} \sum_{j \in \mathrm{TR}} \frac{3}{2}\mathrm{dist}(x_j, V) \leq \frac{3}{2}\mathrm{DIS}.$$

By Lemma 2.1.2, we know that for every $j \in \mathrm{FL}$,

$$\mathrm{dist}(x_j, v) < 12 r_{i-1} \tag{2.2}$$

since $\mathrm{flag}(j)$ remains "False" at the $(i-1)$-th round. Hence

$$\begin{aligned}
\mathrm{DIS} &= \sum_{j \in [n]} \mathrm{dist}(x_j, V) \\
&= \sum_{j \in \mathrm{TR}} \mathrm{dist}(x_j, V) + \sum_{j \in \mathrm{FL}} \mathrm{dist}(x_j, V) \\
&\leq \sum_{j \in \mathrm{TR}} 2\mathrm{dist}(x_j^{(i_j)}, V) + \sum_{j \in \mathrm{FL}} 12 r_{i-1} && \text{(Lemma 2.1.2 and Eq. (2.2))} \\
&= 2D + 24 T r_i && (r_i = r_{i-1}/2) \\
&< 2D + 24D/48 && \text{(Line 3 and the fact that } i \text{ is the last round)} \\
&= \frac{5}{2} D.
\end{aligned}$$

Hence $D \in [\frac{2}{5}\mathrm{DIS}, \frac{3}{2}\mathrm{DIS}]$.

Now we consider the sample complexity. Since $m_{i+1} \geq 4m_i$, the total sample

complexity is bounded by the number of samples taken at the last round of the while-loop. Therefore, we only need to bound r_i at the last round of the while-loop since the number of samples m_i is determined by r_i.

Suppose $i > 1$ and $r_{i-1} < \frac{\mathrm{DIS}}{1000n}$. Then for every $j \in [n]$ such that $\mathrm{dist}(x_j, V) \geq \frac{12\mathrm{DIS}}{500n} \geq 12r_{i-2}$, $\mathrm{flag}(j)$ has been set to be "TRUE" at the $(i-2)$-th round by Lemma 2.1.2. Then in Line 3 of the $(i-1)$-th round, we have

$$\begin{aligned} D &\geq \frac{1}{2} \sum_{j:\mathrm{dist}(x_j,V)\geq \frac{12\mathrm{DIS}}{500n}} \mathrm{dist}(x_j, V) \geq \frac{1}{2}\left(\mathrm{DIS} - \sum_{j:\mathrm{dist}(x_j,V)< \frac{12\mathrm{DIS}}{500n}} \mathrm{dist}(x_j, V)\right) \\ &\geq \frac{1}{2}\left(\mathrm{DIS} - n \cdot \frac{12\mathrm{DIS}}{500n}\right) \geq \frac{1}{2}(1 - \frac{12}{500})\mathrm{DIS} = 0.488\mathrm{DIS}. \end{aligned}$$

However, we have $48Tr_{i-1} \leq 48n \cdot \frac{\mathrm{DIS}}{1000n} = 0.048\mathrm{DIS} < D$. Hence the algorithm terminates in Line 3 of $(i-1)$-th round, which is a contradiction. Therefore, based on $\mathcal{E}$, we have $r_i = r_{i-1}/2 \geq \frac{\mathrm{DIS}}{2000n}$ when the algorithm terminates. On the other hand, note that $r_1 = 1$ (the algorithm may terminate in the first round). Thus, the total sample complexity is upper bounded by

$$\tilde{O}_{\delta^{-1}}(dn \cdot \min\{\frac{\mathrm{DIS}}{n}, 1\}^{-2}) = \tilde{O}_{\delta^{-1}}\left(d(n^3\mathrm{DIS}^{-2} + n)\right).$$

■

2.1.2 Noisy K-Median

We are ready to design an algorithm for the Noisy k-Median problem. The main idea is to estimate the mean of each arm, keeping increasing the accuracy of those estimations and computing an approximated k-median clustering on empirical data points until we have confidence that the current clustering is a good approximation of the optimal clustering. The algorithm uses a constant factor approximation algorithm $\mathsf{BPRS}(\cdot)$ for k-median as a subroutine. In particular, $\mathsf{BPRS}(n, k, d, X)$ is given a set of n points in $\mathbb{R}^d$ and outputs a set of k centers which is an $\alpha < 2.676$ approximation for the k-median on X.

Next, we highlight the critical steps in NoisyKmedian. Our algorithm iteratively estimates every x_j, using more and more samples. In Line 3, the algorithm uses DisNATKP to computes $C_1^{(i)}$, the sum of distances from all arms to the current set A_{i-1} of k-centers. Then in Line 6, the algorithm estimates x_j for each arm by the average $x_j^{(i)}$ of m_i samples. Observe that m_i increases exponentially in each round which implies that the estimation $x_j^{(i)}$ is guaranteed to be more and more accurate. In Lines 7 and 8, the algorithm computes an approximate k-median clustering A_i on estimations $\left\{x_j^{(i)}\right\}_{j\in[n]}$ and compute the approximate k-median value $C_2^{(i)}$. In Line 9, the algorithm checks if the two k-median values $C_1^{(i)}$ and $C_2^{(i)}$ differ by much. If they are close, the algorithm has an evidence that $C_1^{(i)}$ is an $O(1)$-approximate k-median value and terminates.

Algorithm 2 NoisyKmedian$(X = \{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, \delta)$

Require: A sample access $\mathcal{O}(x_i)$ for every $i \in [n]$ and a confidence parameter $\delta \in (0, 1)$.

Ensure: An $O(1)$-approximate k-median clustering on X and an $O(1)$-approximate k-median value..

1: $r_1 \leftarrow 1$, $i \leftarrow 1$, $C_1 \leftarrow \infty$, $A_0 \leftarrow$ a set of arbitrary k centers.
2: **while** TRUE **do**
3: $\quad C_1^{(i)} \leftarrow \mathsf{DisNATKP}(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, A_{i-1}, \frac{\delta}{100i^2})$.
4: $\quad r_i \leftarrow \min\{r_i, \frac{C_1^{(i)}}{80n}\}$
5: $\quad m_i = O\big(r_i^{-2}(d + \log(\delta^{-1}ni))\big)$.
6: $\quad$ For every $j \in [n]$, take m_i samples from $\mathcal{O}(x_j)$ and compute their mean $x_j^{(i)}$ as an estimate of x_j.
7: $\quad$ Set $A_i \leftarrow$BPRS$(n, k, d, X_i := \{x_1^{(i)}, \ldots., x_n^{(i)}\})$.
8: $\quad C_2^{(i)} \leftarrow \sum_{j=1}^n \mathrm{dist}(x_j^{(i)}, A_i)$.
9: $\quad$ **if** $C_1^{(i)}/C_2^{(i)} \leq 10$ **then**
10: $\quad\quad$ BREAK THE WHILE LOOP.
11: $\quad$ **end if**
12: $\quad r_{i+1} \leftarrow r_i/2$, $i \leftarrow i + 1$.
13: **end while**
14: **return** $A_{i-1}, C_1^{(i)}$.

The main theorem is as follows.

Theorem 2.1.3. *With probability at least $1-\delta$, NoisyKmedian(X, δ) outputs an $O(1)$-approximate k-median clustering and an $O(1)$-approximate k-median value on X, with*

sample complexity

$$\tilde{O}_{\delta^{-1}}\bigg(d(n^3\mathrm{OPT}^{-2}+n)\bigg),$$

where $\mathrm{OPT} := \min_{V\subset\mathbb{R}^d,|V|=k}\mathrm{cost}(X,V)$ *is the optimal* k*-median value on* X*.*

Proof. Define event $\mathcal{E}_i = \{\omega : \forall j \in [n], \|x_j^{(i)} - x_j\| \le r_i\}$ and $\mathcal{E}_i' = \{$ The i-th call of DisNATKP succeeds$\}$. Let $\mathcal{E} = \cap_{i\ge 1}(\mathcal{E}_i \cap \mathcal{E}_i')$. Similar to the proof of Theorem 1.4.1, we have $\Pr[\cap_{i\ge1}\mathcal{E}_i] \ge 1-\delta/3$. By Lemma 2.1.1 and the union bound, we have $\Pr[\cap_{i\ge1}\mathcal{E}_i'] \ge 1-\frac{\delta}{100}(1+\frac{1}{2^2}+\frac{1}{3^2}+\ldots) \ge 1-\delta/3$. Combining the above two inequalities, we know that $\Pr[\mathcal{E}] \ge 1-\delta$. Next, we need the following lemma for preparation, and its proof can be found in the Appendix A.

Lemma 2.1.4. *Conditioned on* $\mathcal{E}$*, suppose NoisyKmedian*$(\{\mathcal{O}(x_i) : i = 1,2,\ldots,n\},\delta)$ *terminates at the* i*-th round, then* $r_i \ge \frac{\mathrm{OPT}}{200n}$ *at the end of the algorithm.*

Now we come back to the proof of Theorem 2.1.3. Consider the sample complexity, it is dominated by the number of samples of the last round. Suppose the algorithm terminates at the i-th round. Again, we need to bound r_i. By Lemma 2.1.4, we have $r_i \ge \frac{\mathrm{OPT}}{200n}$. Also note that $r_1 \le 1$. Hence the overall sample complexity is bounded by,

$$\tilde{O}_{\delta^{-1}}(dn\min\{r_i,1\}^{-2}) = \tilde{O}_{\delta^{-1}}(dn^3\mathrm{OPT}^{-2}+n).$$

Finally, we need to prove that the output A_{i-1} and $C_1^{(i)}$ satisfy the theorem. Recall that $C_1^{(i)} = \mathsf{DisNATKP}(\{\mathcal{O}(x_i) : i = 1,2,\ldots,N\}, A_{i-1}, \frac{\delta}{100i^2})$. Conditioned on $\mathcal{E}$, $C_1^{(i)} \in [\frac{2}{5}\mathrm{cost}(X,A_{i-1}), \frac{3}{2}\mathrm{cost}(X,A_{i-1})]$ by Lemma 2.1.1. Hence we have $C_1^{(i)} \ge \frac{2}{5}\mathrm{cost}(X,A_{i-1}) \ge \frac{2}{5}\mathrm{OPT}$. On the other hand, we need to prove $C_1^{(i)} = O(1)\cdot\mathrm{OPT}$. Assume that $\mathrm{cost}(X,A_{i-1}) > 600\mathrm{OPT}$. Then we have $C_1^{(i)} > 240\mathrm{OPT}$.

Let OPT_1 denote the optimal k-median value on X_i. As in the proof of Lemma

2.1.4, we have that $\mathrm{OPT}_1 \le \mathrm{OPT} + nr_i$. Then we have

$$\begin{aligned} C_2^{(i)} =& \mathrm{cost}(X_i, A_i) \\ \le& 3\mathrm{OPT}_1 && \text{(Defn. of BPRS)} \\ \le& 3(\mathrm{OPT} + nr_i) \\ \le& 3\mathrm{OPT} + 3n \cdot \frac{C_1^{(i)}}{80n} && \text{(Line 4)} \\ \le& 3\mathrm{OPT} + 0.04C_1^{(i)}. \end{aligned}$$

Hence

$$\frac{C_1^{(i)}}{C_2^{(i)}} \ge \frac{C_1^{(i)}}{3\mathrm{OPT} + 0.04C_1^{(i)}} \overset{C_1^{(i)} > 240\mathrm{OPT}}{>} \frac{240\mathrm{OPT}}{3\mathrm{OPT} + 10\mathrm{OPT}} > 10,$$

which contradicts the fact that the algorithm terminates at the i-th round. Hence we have $\mathrm{cost}(X, A_{i-1}) \le 600\mathrm{OPT}$. It also implies that $C_1^{(i)} \le \frac{3}{2}\mathrm{cost}(X, A_{i-1}) \le 900\mathrm{OPT}$ which completes the proof.

■

2.2 A Testing-Based Algorithm for $d = k = 1$

Theorem 2.1.3 does not match the sample complexity lower bound in Theorem 2.4.5 on the parameter n. When $n \to \infty$, the upper bound is asymptotically $n^3\mathrm{OPT}^{-2} + n$ (regarding d as constant and ignoring logarithmic factors) while the lower bound (by Theorem 2.4.5)is asymptotically $n^2\mathrm{OPT}^{-2} + n$. This gap exists even in the simplest case $d = k = 1$. In this section, we show how to obtain nearly tight sample complexity upper bound for the simple case of $d = k = 1$. In the next section, we will apply our technique to the general setting.

2.2.1 Testing the Total Distance from n Arms to 1 Arm

We first design a testing-based algorithm CostTester for computing the optimal 1-median clustering in Algorithm 3. Our testing algorithm receives an instance of

multi-armed bandit $\{\mathcal{O}(x_i), i \in [n]\}$, an arm $\mathcal{O}(x)$ as a clustering center, and a number $C > 0$ as input. The algorithm tests whether the sum $\sum_{i=1}^{n} |x_i - x|$ is larger than $10C$ or smaller than $C/10$. Note that the setting of CostTester is slightly different from DisNATKP in Section 2.1.1 since the given center is an uncertain point $\mathcal{O}(x)$ instead of a fixed point. [1] Hence we also need to estimate the location of x in the algorithm.

The main idea is to iteratively set up a threshold C_i (which decreases exponentially), and count the number N_i of data points which are at least C_i-far from the center. Then we use these numbers N_i to construct an estimation of the total distance. If C_i is large, the algorithm estimates the location of each data point and checks whether the distance between each data point to the given point x is larger than C_i (Lines 11-19). If C_i is small, the algorithm cannot afford the number of samples to precisely estimate the location of each data point. To handle this problem, we take some subsamples from arms, work only with the subsampled set, and get an estimation of N_i (Lines 6-10). The algorithm achieves the desired sample complexity by balancing the trade-off between the number of subsampled arms and the number of samples on each sampled arm.

Remark 2.2.1. For the case of $d = 1$, our model is identical to the multi-armed bandit model. Hence we can regard the problem as computing 1-median clustering in the multi-armed bandit model. This problem can be motivated by the following example: given a multi-armed bandit instance, a player suspects that all arms are quite similar and wants to measure the similarity of all arms with as few samples as possible.

Lemma 2.2.2. *Given an instance $X = \{\mathcal{O}(x_i) : i \in [n]\}$, a clustering center $\mathcal{O}(x)$, a cost parameter $C > 0$ and a confidence parameter $\delta \in (0, 1)$, the algorithm CostTester satisfies the following properties:*

- *If* $\mathrm{cost}(X, x) \geq 10C$, CostTester$(X, \mathcal{O}(x), C, \delta)$ *accepts with probability at least* $1 - \delta$.

[1] Note that this slight modification can be considered as a generalization from a fixed center to an uncertain center. The goal of the modification is to simplify the main algorithm in the next subsection.

Algorithm 3 CostTester$(X = \{\mathcal{O}(x_j) : j = 1, 2, \ldots, n\}, \mathcal{O}(x), C, \delta)$

Require: An instance $X = (\{\mathcal{O}(x_j) : j \in [n]\})$, an arm $\mathcal{O}(x)$ as a center, a threshold $C > 0$ and a confidence parameter $\delta \in (0, 1)$.

Ensure: "Accept" or "Reject".

$C_1 \leftarrow \max\{2C, 100\}$, $T \leftarrow 0$.
$L \leftarrow O(\log(C_1 n/C))$ such that $\frac{40C_1 n}{C} \leq 2^L < \frac{80C_1 n}{C}$.
for $i = 1, 2, \ldots, L$ **do**
  $r_i \leftarrow \min\{1, C_i\}$, $m_i \leftarrow O\left(r_i^{-2} \log(nL\delta^{-1})\right)$.
  Take m_i samples from $\mathcal{O}(x)$ and compute their average $x^{(i)}$.
  if $C_i \leq C$ **then**
    Uniformly draw with replacement a subset S_i of size $O\left((\frac{nLr_i}{C})^2 \log(\delta^{-1} nL)\right)$ from X.
    For every $\mathcal{O}(x_j) \in S_i$, take m_i samples from $O(x_j)$ and compute their average $x_j^{(i)}$.
    $N_i \leftarrow \frac{n}{|S_i|}\left|\{\mathcal{O}(x_j) \in S_i : |x^{(i)} - x_j^{(i)}| \geq C_i\}\right|$.
  end if
  if $C_i > C$ **then**
    $N_i \leftarrow 0$.
    for $\mathcal{O}(x_j) \in X$ **do**
      Take m_i samples from $\mathcal{O}(x_j)$ and compute their average $x_j^{(i)}$.
      if $|x^{(i)} - x_j^{(i)}| > C_i$ **then**
        $N_i \leftarrow N_i + 1$.
      end if
    end for
  end if
  $T \leftarrow T + N_i C_i$, $C_{i+1} \leftarrow C_i/2$.
end for
return "Accept" if $T > C$; "Reject" otherwise.

- *If* $\mathrm{cost}(X, x) \leq C/10$, CostTester$(X, \mathcal{O}(x), C, \delta)$ *rejects with probability at least* $1 - \delta$.
- *The sample complexity of* CostTester$(X, \mathcal{O}(x), C, \delta)$ *is* $\tilde{O}\left((n + n^2 C^{-2}) \log^2 \delta^{-1}\right)$.

Proof. We first upper bound the sample complexity of the algorithm. At the i-th round, if $C_i \leq C$, we take

$$m_i |S_i| = O\left((r_i^{-2} \log(nL\delta^{-1}) \cdot (\frac{nLr_i}{C})^2 \log(n\delta^{-1} L)\right) = \tilde{O}\left(n^2 C^{-2} \log^2 \delta^{-1}\right)$$

many samples in this round; meanwhile if $C_i > C$, we take

$$nm_i = O(nr_i^{-2}\log(nL\delta^{-1})) = \tilde{O}((n + nC^{-2})\log\delta^{-1})$$

(noting that $r_i = \min\{1, C_i\}$ and $C_i \le C_1 \in O(\max\{C, 1\})$) many samples at this round.

Note that we have $L = \tilde{O}_{n,C}(1)$ many rounds in total. Therefore, the overall sample complexity is

$$\tilde{O}((n + n^2C^{-2})\log^2\delta^{-1}).$$

So we have proved the third argument.

We now prove the first argument, i.e., if $\mathrm{cost}(X, x) \ge 10C$ then with probability at least $1 - \delta$ the algorithm accepts. We need to prove that with probability at least $1 - \delta$, $T > C$ after L rounds. We first define two events, $\mathcal{E}$ and $\mathcal{E}'$ as follows.

Define events $\mathcal{E}_{ij} = \{\omega : |(x_j^{(i)} - x^{(i)}) - (x_j - x)| \le r_i/2\}$. Note that $x_j^{(i)} - x^{(i)}$ is a Gaussian variable with mean $x_j - x$ and variance $\frac{2}{m_i}$. By the same argument as in the proof of Lemma 2.1.1, we have

$$\Pr[\mathcal{E}_{ij}] \ge 1 - \frac{\delta}{2nL}.$$

Let $\mathcal{E} = \cap\mathcal{E}_{ij}$. Since $i \in [L]$ and $j \in [n]$, by the union bound, we have

$$\Pr[\mathcal{E}] \ge 1 - \frac{\delta}{2}. \tag{2.3}$$

Define the set $M_i = \{j \in [n] : |x_j - x| > 2C_i\}$. We also define $P_i = S_i \cap M_i$ for i satisfying that $C_i \le C$.

For every i satisfying that $C_i \le C$, we define the event $\mathcal{E}_i'$ to be the inequality $\left|\frac{n}{|S_i|}|P_i| - |M_i|\right| \le CL^{-1}C_i^{-1}/8$, and $\mathcal{E}' = \cap_{i:C_i\le C}\mathcal{E}_i'$. By Theorem 1.4.2, we have that

$$\Pr\left(\left|\frac{1}{|S_i|}|P_i| - \frac{1}{n}|M_i|\right| > \varepsilon_i\right) \le 2\exp^{-2|S_i|\varepsilon_i^2}$$

for $\varepsilon_i = \frac{C}{8nLC_i}$.

Since $|S_i| = O(n^2L^2C_i^2C^{-2}\log(\delta^{-1}L))$, we have that,

$$\Pr[\mathcal{E}_i'] = 1 - \Pr\left(\left|\frac{n}{|S_i|}|P_i| - |M_i|\right| > CL^{-1}C_i^{-1}/8\right) \geq 1 - \frac{\delta}{2L}.$$

By the union bound, we have $\Pr[\mathcal{E}'] \geq 1 - \frac{\delta}{2}$. Combining with Inequality (2.3), we have $\Pr[\mathcal{E} \cap \mathcal{E}'] \geq 1 - \delta$ by the union bound.

Conditioned on $\mathcal{E} \cap \mathcal{E}'$, if $C_i \leq C$ and $j \in P_i$ then $|x_j^{(i)} - x^{(i)}| \geq |x_j - x| - |x_j^{(i)} - x^{(i)} - (x_j - x)| \geq 2C_i - C_i = C_i$ by the triangle inequality. Hence, such entry j must be counted in Line 9, i.e.,

$$N_i \geq \frac{n}{|S_i|}|P_i| \overset{\mathcal{E}'}{\geq} |M_i| - CL^{-1}C_i^{-1}/8.$$

On the other hand, if $C > C_i$, since the algorithm checks every $x_j, j \in [n]$ we know that if $j \in M_i$ then j contributes one to N_i which implies that $N_i \geq |M_i|$. Therefore, we conclude that

$$N_i \geq |M_i| - CL^{-1}C_i^{-1}/8 \tag{2.4}$$

for every i, on $\mathcal{E} \cap \mathcal{E}'$.

We only need to prove that $T > C$ conditioned on $\mathcal{E} \cap \mathcal{E}'$. As a consequence, the algorithm accepts with probability at least $1-\delta$. We consider the following two cases.

1) If there is some j such that $|x_j - x| > 3C$, then for some $i \in [L]$ satisfying that $C < C_i \leq 2C$, we have

$$|x_j^{(i)} - x^{(i)}| \overset{\text{triangle ineq.}}{\geq} |x_j - x| - |x_j^{(i)} - x^{(i)} - (x_j - x)| \overset{\mathcal{E}}{>} 3C - 2C = C.$$

Hence at the i-th round, $N_i \geq 1$ which implies $T \geq C_i > C$.

2) Assume that $|x_j - x| \leq 3C$ for every $j \in [n]$. In this case, we need the following lemma.

Lemma 2.2.3. *Assume that $|x_j - x| \leq 3C$ for every $j \in [n]$. Conditioned on $\mathcal{E} \cap \mathcal{E}'$, we have that*

$$\text{cost}(X, x) \leq 2 \sum_{i=1}^{L} C_i |M_i| + \frac{C}{10}.$$

Proof. For every $j \in [n]$, let α_{ij} denote the indicator function of $j \in M_i$. Since $|x_j - x| \leq 3C < 2C_1$ which implies that $\alpha_{1j} = 0$, there must exist some $i^* \in \{1, 2, \ldots, L\}$ which is the smallest number such that $\alpha_{ij} = 0$. By the definition of M_i and the fact that $C_{i+1} = C_i/2$, we know that,

$$|x_j - x| \leq 2C_{i_*} = 2(C_L + C_L + C_{L-1} + C_{L-2} + \ldots + C_{i^*-1}) = 2C_L + \sum_{i=1}^{L} 2C_i \alpha_{ij}.$$

Moreover, $|M_i| = \alpha_{i1} + \ldots + \alpha_{in}$. So we have

$$\text{cost}(X, x) = \sum_{i=1}^{n} |x_j - x| \leq \sum_{j=1}^{n} \left(2C_L + \sum_{i=1}^{L} 2C_i \alpha_{ij} \right) = 2nC_L + 2 \sum_{i=1}^{L} C_i |M_i| \leq \frac{C}{10} + 2 \sum_{i=1}^{L} C_i |M_i|,$$

where the last inequality is due to the fact that $C_L = \frac{C_1}{2^{L-1}} \leq \frac{C_1}{20 C_1 n / C} = \frac{C}{20n}$. ■

By Lemma 2.2.3, we have that,

$$\begin{aligned}
T =& \sum_{i=1}^{L} N_i C_i \\
\geq& \sum_{i=1}^{L} (|M_i| - CL^{-1} C_i^{-1}/8) C_i && \text{(Ineq. (2.4))} \\
=& \sum_{i=1}^{L} |M_i| C_i - C/8 \\
\geq& \text{cost}(X, x)/2 - C/20 - C/8 && \text{(Lemma 2.2.3)} \\
\geq& 5C - C/20 - C/8 && (\text{cost}(X, x) \geq 10C) \\
>& C.
\end{aligned}$$

Finally, we prove the second argument, i.e., if $\text{cost}(X, x) \leq C/10$, the algorithm rejects with probability at least $1 - \delta$. Define sets $R_i = \{j \in [n] : |x_j - x| \geq C_i/2\}$, $Q_i = S_i \cap R_i$, and events $\mathcal{E}_i'' = \{\omega : \left| \frac{n}{|S_i|} |Q_i| - |R_i| \right| \geq CL^{-1} C_i^{-1}/8\}$ and $\mathcal{E}'' = \cap \mathcal{E}_i''$. Similar to the argument of $\mathcal{E}'$, we can prove that $\Pr[\mathcal{E}''] \geq 1 - \delta/2$ and $\Pr[\mathcal{E} \cap \mathcal{E}''] \geq 1 - \delta$.

To prove our second argument, we only need to show that if $\mathrm{cost}(X, x) \le C/10$, then $T < C$ conditioned on $\mathcal{E} \cap \mathcal{E}''$.

Conditioned on $\mathcal{E} \cap \mathcal{E}''$, if $C_i \le C$, for any $j \notin Q_i$, we have

$$|x_j^{(i)} - x^{(i)}| \overset{\text{triangle ineq.}}{\le} |x_j - x| + |(x_j^{(i)} - x^{(i)}) - (x_j - x)| \overset{j \notin Q_i \text{ and } \mathcal{E}}{<} C_i/2 + r_i/2 \overset{r_i \le C_i}{\le} C_i/2 + C_i/2 = C_i.$$

It implies that $x_j \notin \{\mathcal{O}(x_l) \in S_i : |x^{(i)} - x_l^{(i)}| \ge C_i\}$. Hence we have $N_i \le \frac{n}{|S_i|}|Q_i|$ in Line 9. Consequently,

$$N_i \le \frac{n}{|S_i|}|Q_i| \overset{\mathcal{E}''}{\le} |R_i| + CL^{-1}C_i^{-1}/8.$$

On the other hand, consider the case that $C_i > C$. By the same argument as the previous case, each $j \notin R_i$ satisfies that $|x_j^{(i)} - x^{(i)}| < C_i$. Hence $N_i \le |R_i|$ in this case. So we conclude that, for every $i \in [L]$,

$$N_i \le |R_i| + CL^{-1}C_i^{-1}/8. \tag{2.5}$$

Next, we need the following lemma which is similar to Lemma 2.2.3. The proof can be found in Appendix A.

Lemma 2.2.4. *Conditioned on $\mathcal{E} \cap \mathcal{E}''$, we have*

$$\mathrm{cost}(X, x) \ge \frac{1}{2}\sum_{i=1}^{L} C_i|R_i|.$$

By Lemma 2.2.4, we have that,

$$\begin{aligned}
T =& \sum_{i=1}^{L} N_i C_i \\
\leq& \sum_{i=1}^{L} (|R_i| + CL^{-1}C_i^{-1}/8)C_i && \text{(Ineq. (2.5))} \\
=& \sum_{i=1}^{L} |R_i|C_i + C/8 \\
\leq& 2\text{cost}(X, x) + C/8 && \text{(Lemma 2.2.4)} \\
\leq& C/5 + C/8 && (\text{cost}(X, x) \leq C/10) \\
<& C.
\end{aligned}$$

■

2.2.2 Noisy 1-Median

Now we show how to compute noisy 1-median using Algorithm 3. Our main idea is to first sample enough arms such that there exists an $O(1)$-approximate center among sampled arms. Then we design a binary search algorithm to compute the optimal one among the sampled arms based on Algorithm CostTester, see Algorithm 4. The estimation for the optimal arm is exactly an $O(1)$-approximate center.

Before analyzing Noisy1median, we first have the following lemma which shows that $O(\log \delta^{-1})$ sampled arms in Line 1 of Noisy1median is enough.

Lemma 2.2.5. *Let $X = \{\mathcal{O}(x_1), \ldots, \mathcal{O}(x_n)\}$. If S is a uniformly i.i.d. sample of size $O(\log \delta^{-1})$ from X, then with probability at least $1 - \delta$, there exists $\mathcal{O}(x) \in S$ such that x is a 3-approximate 1-median center on X.*

Proof. Let y be the optimal 1-median center on X and let $\text{OPT} = \sum_{i=1}^{n} |y - x_i|$.

W.l.o.g., we assume that

$$|x_1 - y| \leq |x_2 - y| \leq \ldots \leq |x_n - y|. \tag{2.6}$$

Algorithm 4 Noisy1median$(X = \{\mathcal{O}(x_j) : j = 1, 2, \ldots, n\}, \delta)$

Require: An instance $X = (\{\mathcal{O}(x_j) : j \in [n]\})$ and a confidence parameter $\delta \in (0, 1)$.
Ensure: An $O(1)$-approximate 1-median clustering on X and an $O(1)$-approximate 1-median value.

1: Uniformly draw with replacement a subset $S = \{\mathcal{O}(x_{\alpha_1}), \ldots, \mathcal{O}(x_{\alpha_m})\}$ of size $m = O(\log \delta^{-1})$ from X.
2: $C = 100n$.
3: **if** $\forall l \in [m]$ CostTester$(X, \mathcal{O}(x_{\alpha_l}), C, \frac{\delta}{20m})$ accepts **then**
4: For every $j \in [n]$, take $O(\log(n\delta^{-1}))$ many samples from arm $\mathcal{O}(x_j)$ and compute their average x'_j.
5: Compute the optimal 1-median center x on $X' = \{x'_1, x'_2, \ldots, x'_n\}$ by computing their median.
6: **return** x and $\text{cost}(X', x)$.
7: **end if**
8: $i \leftarrow 1$, $j \leftarrow \alpha_l$ satisfying that CostTester$(X, \mathcal{O}(x_{\alpha_l}), C, \frac{\delta}{20m})$ rejects in Line 3 (breaking ties arbitrarily).
9: **while** TRUE **do**
10: **if** $\forall l \in [m]$ CostTester$(X, \mathcal{O}(x_{\alpha_l}), C, \frac{\delta}{10mi(i+1)})$ accepts **then**
11: Take $O(n^2 C^{-2} \log \delta^{-1})$ samples from $\mathcal{O}(x_j)$ and return their average x and C.
12: **end if**
13: $j \leftarrow \alpha_l$ satisfying that CostTester$(X, \mathcal{O}(x_{\alpha_l}), C, \frac{\delta}{10mi(i+1)})$ rejects in Line 10 (breaking ties arbitrarily).
14: $C \leftarrow C/10$, $i \leftarrow i + 1$.
15: **end while**

So the probability that T contains some $\mathcal{O}(x_j)$ for $j \leq \frac{n}{10}$ is at least

$$1 - 0.9^{|T|} \geq 1 - \delta.$$

Suppose $\mathcal{O}(x_j) \in S$ for some $j \leq \frac{n}{10}$. We finish the proof by noting that

$$\begin{aligned}
\sum_{l=1}^{n} |x_j - x_l| \leq & \sum_{l=1}^{n} (|y - x_l| + |y - x_j|) && \text{(triangle ineq.)} \\
= & \mathrm{OPT} + n|y - x_j| \\
\leq & \mathrm{OPT} + \frac{n}{n-j} \sum_{l=j+1}^{n} |y - x_l| && \text{(Ineq. (2.6))} \\
\leq & \mathrm{OPT} + \frac{n}{n-j} \mathrm{OPT} \\
\leq & 3\mathrm{OPT}. && (j \leq \frac{n}{10})
\end{aligned}$$

■

Now we are ready to prove the correctness and sample complexity of Algorithm Noisy1median. The main theorem is as follows. Note that by the following theorem and Theorem 2.4.1, we achieve the nearly tight sample complexity for obtaining $O(1)$-approximate 1-median value in 1-dimension.

Theorem 2.2.6. *Given an instance $X = \{\mathcal{O}(x_j) : j \in [n]\}$ and a confidence parameter $\delta \in (0, 1)$. With probability at least $1-\delta$, the algorithm Noisy1median(X, δ) returns an $O(1)$-approximate 1-median center and an $O(1)$-approximate 1-median value on X, with sample complexity*

$$\tilde{O}_{\delta^{-1}}(n + n^2 \mathrm{OPT}^{-2})$$

where $\mathrm{OPT} = \min_{y \in \mathbb{R}} \mathrm{cost}(X, y)$ is the optimal 1-median value.

Proof. We define event $\mathcal{E}$ that every call of CostTester succeeds and there is a 3-approximate 1-median center in T. By Lemmas 2.2.5 and 2.3.1 and the union bound, we have

$$\Pr[\mathcal{E}] \geq 1 - m \cdot \left(\frac{\delta}{20m} + \sum_{i=1}^{+\infty} \frac{\delta}{10mi(i+1)} \right) - \frac{\delta}{4} \geq 1 - \delta/2.$$

We condition on $\mathcal{E}$ in what follows. For preparation, we have the following lemma.

Lemma 2.2.7. *Conditioned on $\mathcal{E}$, if every CostTester in Line 3 (or Line 10) accepts, then $\mathrm{OPT} \geq C/30$ at that iteration. If $\mathrm{OPT} > 10C$ in Line 3 (or Line 10), then*

every CostTester *at that iteration accepts.*

Proof. For the first argument, if $\mathrm{OPT} < C/30$, by Lemma 2.2.5 there is a $x_{\alpha_l} \in S$ such that $\mathrm{cost}(X, x_{\alpha_l}) \leq 3\mathrm{OPT} < C/10$. It means that $\mathsf{CostTester}(X, \mathcal{O}(x_{\alpha_l}), C, \delta')$ rejects by Lemma 2.2.2. The second argument follows from Lemma 2.2.2 since $\mathrm{cost}(X, x_{\alpha_l}) \geq \mathrm{OPT} > 10C$ for all $l \in [m]$. ■

Now we come back to the proof of Theorem 2.2.6. We consider the first "IF" sentence in Line 3. If all CostTester accept, we have $\mathrm{OPT} > \frac{C}{30} = \frac{10n}{3}$ by Lemma 2.2.7. Define by event $\mathcal{E}'$ the algorithm estimates every arm x_j by an empirical mean x'_j such that $|x_j - x'_j| \leq \frac{1}{3}$. By union bound, the probability of $\mathcal{E}'$ is at least $1 - n \cdot \frac{\delta}{2n} = 1 - \delta/2$. Conditioned on $\mathcal{E} \cap \mathcal{E}'$, we argue that the optimal 1-median center x on $X' := \{x'_1, x'_2, \ldots, x'_n\}$ is an $O(1)$-approximate 1-median center on $X = \{x_1, \ldots, x_n\}$. Assume the optimal center on X is x^*. We have the following inequality

$$
\begin{aligned}
\mathrm{cost}(X, x) \leq & \mathrm{cost}(X', x) + \sum_{i=1}^{n} |x_j - x'_j| && \text{(triangle ineq.)} \\
\leq & \mathrm{cost}(X', x^*) + \sum_{i=1}^{n} |x_j - x'_j| && \text{(Defn. of } x\text{)} \\
\leq & \mathrm{cost}(X, x^*) + \sum_{i=1}^{n} |x_j - x'_j| + \sum_{i=1}^{n} |x_j - x'_j| && \text{(triangle ineq.)} \\
\leq & \mathrm{OPT} + \frac{2n}{3} && (\mathcal{E}') \\
\leq & \mathrm{OPT} + 0.2\mathrm{OPT} && (\mathrm{OPT} > \frac{10n}{3}) \\
= & 1.2\mathrm{OPT}.
\end{aligned}
$$

By the same argument, we can prove that $0.8\mathrm{OPT} \leq \mathrm{cost}(X', x) \leq 1.2\mathrm{OPT}$. Hence, the output $\mathrm{cost}(X', x)$ is an $O(1)$-approximation of OPT. The success probability is at least $\Pr[\mathcal{E} \cap \mathcal{E}'] \geq 1 - \delta$. Moreover, the sample complexity is

$$\tilde{O}_{\delta^{-1}}(n) = \tilde{O}_{\delta^{-1}}(n + n^2\mathrm{OPT}^{-2})$$

since $\mathrm{OPT} > \frac{10n}{3}$.

If there is some tester CostTester in Line 3 rejects, the algorithm will enter the while-loop in Line 9. Consider the sample complexity. It is again dominated by the number of samples of the last round since C decreases exponentially in each round. By Lemma 2.2.7,we know that the algorithm terminates once $C \leq \mathrm{OPT}/10$. Therefore, we have $C \geq \mathrm{OPT}/100$ when the algorithm terminates, by the updating rule in Line 14. Since T contains only $\tilde{O}_{\delta^{-1}}(1)$ sample access, it implies that the sample complexity is bounded by

$$\tilde{O}_{\delta^{-1}}(n^2\mathrm{OPT}^{-2} + n),$$

by Lemma 2.2.2 and Line 11.

For the correctness, by Lemma 2.2.7, we know that $\mathrm{OPT} \geq C/30$ when the algorithm terminates. Hence C is an $O(1)$-approximation of OPT. Note that in Line 11, x_j is set to be a center such that CostTester rejects for the parameter $10C$ in the previous round. Then by Lemma 2.2.2, we have $\mathrm{cost}(X, x_j) \leq 100C \leq 3000\mathrm{OPT}$. Hence x_j is an $O(1)$-approximate 1-median center on X. By Line 11, the algorithm outputs x which is the average of $m = O(n^2C^{-2}\log\delta^{-1})$ samples from $\mathcal{O}(x_j)$. Hence $\sqrt{m}x$ follows from the Gaussian distribution $N(\sqrt{m}x_j, I_d)$. By union bound, we have that $|x_j - x| \leq C/n$ with probability at least $1 - \delta/2$. Then we have

$$\mathrm{cost}(X, x) \overset{\text{triangle ineq.}}{\leq} \mathrm{cost}(X, x_j) + n|x - x_j| \leq O(1)\cdot\mathrm{OPT} + C \overset{C\leq 30\mathrm{OPT}}{\leq} O(1)\cdot\mathrm{OPT}.$$

Since $\Pr[\mathcal{E}] \geq 1 - \delta/2$, the algorithm successes with probability at least $1 - \delta$. ■

2.3 A Testing-Based Algorithm for General Case: Sharper Dependence on n

In the previous section, we present a testing-based algorithm for computing the 1-median center/value in 1-dimension with nearly optimal sample complexity. In this section, we extend this technique to the general setting.

The extension to d-dimension is straightforward. We only need a different concentration inequality. However, to extend to k centers is much harder since we do not have a similar lemma as Lemma 2.2.5. A simple idea is to regard every arm as a candidate of centers and estimate each of them to an error of OPT/n. However, the sample complexity is still as large as $n^3\mathrm{OPT}^{-2}$. To overcome this problem, we can not consider all arms. Instead, we iteratively subsample arms and estimate the location of subsampled arms with different sample numbers in different iterations. We will show that the collection of estimations contain a good approximate k-median clustering. Our approach to solving this issue is somewhat related to the "Chaining method" (see, e.g., [78]).

2.3.1 Testing the Total Distance from n Arms to k Centers

We first present an algorithm TestNATKP which is very similar to algorithm CostTester except that the parameters d and k are generalized. The proof of Lemma 2.3.1 is almost identical to the proof of Lemma 2.2.2 and can be found in Appendix A. Note that the sample complexity in Lemma 2.3.1 saves a parameter n compared to that of Algorithm DisNATKP.

Lemma 2.3.1. *The following holds:*

- *If* $\mathrm{cost}(X, V) > 10C$ *then* TestNATKP(X, V, C, δ) *accepts with probability at least* $1 - \delta$.
- *If* $\mathrm{cost}(X, V) < C/10$ *then* TestNATKP(X, V, C, δ) *rejects with probability at least* $1 - \delta$.
- *The sample complexity of* TestNATKP(X, V, C, δ) *is* $\tilde{O}(d(n + n^2C^{-2}) \log^2 \delta^{-1})$.

2.3.2 Testing the Optimal k-Median Value

We now present Algorithm TestKemdian which tests the cost of optimal k-median on X. It is the key algorithm in this section and can be translated to a binary search algorithm for noisy k-median in the next subsection.

Algorithm 5 TestNATKP($X = \{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, V, C, \delta$)

Require: A sample access $\mathcal{O}(x_i)$ to $N(x_i, I_d)$ for every $i \in [n]$, a set V of k points, a threshold $C > 0$, and a confidence parameter $\delta \in (0, 1)$.

Ensure: "Accept" if $\text{cost}(X, V) > 10C$; "Reject" if $\text{cost}(X, V) < C/10$ where $X := \{x_1, \ldots, x_n\}$.

$C_1 \leftarrow \max\{2C, 100\}$, $T \leftarrow 0$.
$L \leftarrow O(\log(C_1 n/C))$ such that $\frac{40C_1 n}{C} \le 2^L < \frac{80C_1 n}{C}$.
for $i = 1, 2, \ldots, L$ **do**
  $N_i \leftarrow 0$.
  Set $r_i \leftarrow \min\{1, C_i\}$, $m_i \leftarrow O(r_i^{-2}(d + \log(nL\delta^{-1})))$.
  if $C_i \le C$ **then**
    Uniformly draw with replacement a subset $S_i \subset [n]$ of size $O(n^2 L^2 r_i^2 C^{-2} \log(nL\delta^{-1}))$.
    For every $j \in S_i$, take m_i samples from $\mathcal{O}(x_j)$ and obtain their average $x_j^{(i)}$ as an estimate of x_j.
    Compute $N_i = \frac{n}{|S_i|}\left|\{j \in S_i : \text{dist}(x_j^{(i)}, V) \ge C_i\}\right|$.
  end if
  if $C_i > C$ **then**
    for $j = 1, 2, \ldots, n$ **do**
      Take m_i samples from $\mathcal{O}(x_j)$ and compute their average $x_j^{(i)}$ as an estimate of x_j.
      if $\min_{v \in V} \|x_j^{(i)} - v\| \ge C_i$ **then**
        $N_i \leftarrow N_i + 1$.
      end if
    end for
  end if
  $T \leftarrow T + N_i C_i$, $C_{i+1} \leftarrow C_i/2$.
end for
return "Accept" if $T > C$ otherwise "Reject".

The key idea is the following: Consider the optimal k-median clustering A for $X = \{\mathcal{O}(x_i) : i \in [n]\}$ and partition every data point x_i into the closest center $a \in A$. This process partitions X into k clusters $X_1, \ldots, X_k$. We want to compute the center of each cluster X_i. Similar to Noisy1median, we still sample arms to include these candidate centers. Taking X_1 as an example, our approach is based on the following facts. If the size of X_1 is large, we only need a small number of sampled arms to include a precise approximation of its center by Lemma 2.2.5. If the size of X_1 is small, we need many sampled arms to include a precise approximation of its center. However, a rough estimation of the location of this center is already enough since few points

are clustered to it. By the above facts, we sample arms iteratively, i.e., we sample m_i (which increases exponentially) arms at the i-th round. If m_i is small, we can only expect that the sampled arms contain approximate centers for large clusters. To estimate the locations of these subsampled arms, we need a proper number of samples taken from each sampled arm. If m_i is large, we can expect that the sampled arms contain approximate centers for small clusters. Then we only need to take few samples from these arms and obtain rough estimations of their locations. The key problem is to balance the trade-off between the number of sampled arms (which are regarded as estimations of optimal centers) and the number of samples taken from each arm, in each iteration.

Algorithm 6 TestKmedian$(X = \{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, k, C, \delta)$

Require: A sample access $\mathcal{O}(x_i)$ for every $i \in [n]$, the number of centers k, a threshold $C > 0$ and a confidence parameter $\delta \in (0, 1)$.

Ensure: "Accept" if the optimal k-median value on X is larger than $10C$. "Reject" and a set of k centers if the optimal k-median value on X is smaller than $C/50$.

1: **for** $i = 0, 1, 2, \ldots, \lceil \log_2 n \rceil$ **do**
2: $n_i \leftarrow O(\frac{n}{2^i} \log(\delta^{-1} n))$, $r_i \leftarrow \frac{C}{100k2^i}$.
3: $m_i \leftarrow O(r_i^{-2}(d + \log(\delta^{-1} i^2 n_i)))$
4: Uniformly draw with replacement a sample S_i of size n_i from $[n]$.
5: For every $j \in S_i$, take m_i samples from $\mathcal{O}(x_j)$ and compute their mean $x_j^{(i)}$ as an estimate of x_j.
6: **end for**
7: $S \leftarrow \{x_j^{(i)} : i = 1, 2, .., \lceil \log_2 n \rceil,\ j \in S_i\}$.
8: **return** "Reject" and V if there is a size-k set $V \subset S$ satisfying that TestNATKP$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, V, C, \frac{\delta}{|S|^k})$ [2]. Here, for all size-k sets $V \subset S$, we use the same random samples when calling $\mathcal{O}(x_i)$ $(i \in [n])$ in TestNATKP$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, V, C, \frac{\delta}{3|S|^k})$. Otherwise "Accept".

The main lemma is as follows.

Lemma 2.3.2. *TestKmedian(X, k, C, δ) satisfies the following:*

- *If the optimal k-median value on X is larger than $10C$, the algorithm accepts with probability at least $1 - \delta$.*

[2] If there are many such sets V, output an arbitrary one.

- *If the optimal k-median value on X is smaller than $C/100$, the algorithm rejects and outputs a size-k set V satisfying that $\mathrm{cost}(X, V) \le 10C$, with probability at least $1 - \delta$.*

- *The sample complexity is $\tilde{O}_{\delta^{-1}}(d(k^2 n^2 \mathrm{C}^{-2} + k^2 n))$.*

Proof. Let OPT denote the optimal k-median value on X. Define an event $\mathcal{E}_1 = \{\omega : \forall V \subset S, |V| = k$ $\mathsf{TestNATKP}(\{\mathcal{O}(x_i) : i \in [n]\}, V, C, \frac{\delta}{3|S|^k})$ succeeds$\}$. By Lemma 2.3.1 and the union bound, we know that

$$\Pr[\mathcal{E}_1] \ge 1 - \binom{|S|}{k} \cdot \frac{\delta}{3|S|^k} \ge 1 - \delta/3.$$

For the first argument, assume OPT $> 10C$. Then for every $V \subset S, |V| = k$ we have $\mathrm{cost}(X, V) > 10C$. So conditioned on $\mathcal{E}_1$, we know that every $\mathsf{TestNATKP}(\{\mathcal{O}(x_i) : i \in [n]\}, V, C, \frac{\delta}{|S|^k})$ accepts. Hence $\mathsf{TestKmedian}$ accepts in this case.

For the second argument, assume OPT $< C/100$. Let $V^* = \mathrm{argmin}_{V \in \mathbb{R}^d : |V| = k} \mathrm{cost}(X, V)$ and write $V^* = \{v_1, v_2, \ldots, v_k\}$. Let

$$V_i = \{x_j \in X : \mathrm{dist}(x_j, V) = \|x_j - v_i\|\},$$

i.e., V_i is the collection of points in X which is clustered to v_i (breaking ties arbitrarily).

Next, we present the following generalization of Lemma 2.2.5. The proof is almost identical and can be found in Appendix A.

Lemma 2.3.3. *Assume $2^l \le V_i < 2^{l+1}$ for some $l \in \mathcal{N}$. Let A_i be a uniform sample of size $\Theta(\frac{n \log \delta^{-1}}{2^l})$ from $[n]$. Then with probability at least $1 - \delta$, there exists a $j \in A_i$ such that*

$$\mathrm{cost}(V_i, x_j) \le 6\mathrm{cost}(V_i, v_i).$$

Now we come back to the proof of Lemma 2.3.2. For every $i \in [k]$, let l_i be the integer such that $2^{l_i} \le |V_i| < 2^{l_i + 1}$. Define event $\mathcal{E}_2^i = \{\omega : \exists j \in S_{l_i} : \mathrm{cost}(V_i, x_j\}) \le$

$6\text{cost}(V_i, v_i)\}$ and let $\mathcal{E}_2 = \cup_{i=1}^{k} \mathcal{E}_2^i$. By the choice of n_i and Lemma 2.3.3, we have that

$$\Pr[\mathcal{E}_2] \geq 1 - n \cdot \frac{\delta}{3n} = 1 - \delta/3.$$

We also define another event $\mathcal{E}_3 = \{\omega : \forall i = 1, 2, \ldots, \lceil \log_2 n \rceil, j \in S_i, \|x_j^{(i)} - x_j\| \leq r_i\}$. By Theorem 1.4.1 and the union bound, we know that

$$\Pr[\mathcal{E}_3] \geq 1 - \sum_{i=1}^{\log n} n_i \cdot \frac{\delta}{3i^2 n_i} \geq 1 - \delta/3.$$

Let $\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$. Then $\Pr[\mathcal{E}] \geq 1 - \delta$ by the union bound. It remains to prove that conditioned on $\mathcal{E}$, the algorithm rejects.

For every $v_i \in V^*$, let $j_i \in S_{l_i}$ denote the index such that $\text{cost}(V_i, x_{j_i}) \leq 6\text{cost}(V_i, v_i)\}$. ($j_i$ must exist since we condition on $\mathcal{E}$.)

Now we show that $P := \{x_{j_i}^{(l_i)} : i = 1, 2, \ldots, k\}$ is indeed a constant factor approximate k-median clustering. Indeed,

$$\begin{aligned}
\text{cost}(X, P) \leq & \sum_{i=1}^{k} \text{cost}(V_i, x_{j_i}^{(l_i)}) && \text{(Defn. of cost)} \\
\leq & \sum_{i=1}^{k} \text{cost}(V_i, x_{j_i}) + |V_i| \|x_{j_i} - x_{j_i}^{(l_i)}\| && \text{(triangle ineq.)} \\
\leq & 6 \sum_{i=1}^{k} \text{cost}(V_i, v_i) + |V_i| \cdot r_{l_i} && \text{(Defn. of } x_{j_i} \text{ and } \mathcal{E}) \\
\leq & 6\text{OPT} + \sum_{i=1}^{k} 2^{l_i+1} \cdot \frac{C}{100k2^{l_i}} && (|V_i| \leq 2^{l_i+1} \text{ and Defn. of } r_{l_i}) \\
\leq & 6C/100 + \frac{C}{50} && (\text{OPT} < C/100) \\
= & \frac{4}{50} C < C/10.
\end{aligned}$$

Recall that we condition on $\mathcal{E}$. Hence $\mathsf{TestNATKP}(\{\mathcal{O}(x_i) : i \in [n]\}, P, C, \frac{\delta}{|S|^k})$ rejects by Lemma 2.3.2. So we conclude that $\mathsf{TestKmedian}$ rejects when $\text{OPT} < C/100$. Moreover, note that the output V satisfies that $\mathsf{TestNATKP}(\{\mathcal{O}(x_i) : i \in [n]\}, V, C, \frac{\delta}{|S|^k})$ rejects. By Lemma 2.3.2, we have $\text{cost}(X, V) \leq 10C$.

Finally, we consider the sample complexity. There are two places involving taking samples from the oracles: Line 5 and Line 8. For Line 5, the sample complexity is upper bounded by

$$\tilde{O}_{\delta^{-1}}(\sum_{i=1}^{\lceil \log_2 n \rceil} n_i r_i^{-2} d) = \tilde{O}_{\delta^{-1}}(\sum_{i=1}^{\lceil \log_2 n \rceil} k^2 n 2^i C^{-2} d) = \tilde{O}_{\delta^{-1}}(dk^2 n^2 C^{-2}).$$

For Line 8, the sample complexity is bounded by a single call of $\mathsf{TestNATKP}(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, V, C, \frac{\delta}{|S|^k})$ since we use the same samples over all V. By Lemma 2.3.2, the sample complexity is

$$\tilde{O}(d(n^2 C^{-2} + n) \log^2 |S|^k) = \tilde{O}_{\delta^{-1}}(dk^2(n^2 C^{-2} + n)).$$

So the total sample complexity is upper bounded by

$$\tilde{O}_{\delta^{-1}}(dk^2(n^2 \mathrm{C}^{-2} + n)).$$

■

2.3.3 Noisy k-Median

Now we are ready to design a simple binary search algorithm based on TestKmedian. The idea is straightforward – to guess the optimal k-median value iteratively and apply TestKmedian to test the guess.

We summarize the main theorem as follows. The proof is very similar to the proof of Theorem 2.2.6 and can be found in Appendix A.

Theorem 2.3.4. *With probability at least $1 - \delta$, NoisyKmedian2(X, k, δ) outputs an $O(1)$-approximate k-median clustering and an $O(1)$-approximate k-median value on X, with sample complexity*

$$\tilde{O}_{\delta^{-1}}(dk^2(n^2 \mathrm{OPT}^{-2} + n))$$

Algorithm 7 NoisyKmedian2$(X = \{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, k, \delta)$

Require: A sample access $\mathcal{O}(x_i)$ for every $i \in [n]$, the number of centers k, a confidence parameter $\delta \in (0, 1)$.

Ensure: An $O(1)$-approximate k-median clustering on X and an $O(1)$-approximate k-median value.

1: $C \leftarrow 1000n$, $i \leftarrow 1$.
2: If TestKmedian$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, k, C, \delta/20)$ accepts then
3: **return** NoisyKmedian$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n]\}, \delta/10)$.
4: **while** TRUE **do**
5: **if** TestKmedian$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, k, C, \frac{\delta}{10i(i+1)})$ rejects **then**
6: **return** V which is the output of TestKNATKP$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, k, C, \frac{\delta}{10i(i+1)})$ and C.
7: **end if**
8: $C \leftarrow C/10$, $i \leftarrow i + 1$.
9: **end while**

where OPT *is the optimal k-median value on X.*

2.4 Lower Bound

We discuss the sampling lower bound of the noisy k-median problem in this section. We will consider both the instance lower bound and the worst-case lower bound.

2.4.1 Instance Lower Bound

We first consider the instance lower bound for the noisy k-median problem. Recall that in a noisy k-median instance, we are given sample access to n uncertain points $\{\mathcal{O}(x_i) : i \in [n]\}$. The following theorem gives an instance sampling lower bound for the noisy 1-median clustering which shows the tightness of Theorem 2.2.6.

Theorem 2.4.1. *For any noisy 1-median instance in $\mathbb{R}^d$, assume that the optimal 1-median value is* OPT*. Then any $(1-\delta)$-correct algorithm with approximation factor 2 takes at least $\Omega(n^2 \mathrm{OPT}^{-2} \ln \delta^{-1} + n)$ samples in expectation.*

Proof. Assume the n Gaussian distributions for $\{\mathcal{O}(x_i) : i \in [n]\}$ are $N(x_i, I_d), i \in [n]$. W.l.o.g., we assume that the optimal solution of the given instance is $v = (0, 0, \ldots, 0)$.

We first show an $\Omega(n)$ lower bound, which is relatively easy. Suppose for contradiction that a $(1-\delta)$-correct algorithm $\mathbb{A}$ takes $o(n)$ samples in expectation. Now, suppose we pick an arm x_i uniformly at random, and replace x_i with an arm a with center extremely far from v. Then clearly OPT would be dominated by $\|a\|$ and thus change dramatically. But $\mathbb{A}$ can only detect this change with $o(1)$ probability, since it can touch at most $o(n)$ arms. Therefore, $\mathbb{A}$ cannot be $(1-\delta)$-correct with $o(n)$ samples.

Next we prove an $\Omega(n^2\mathrm{OPT}^{-2}\ln\delta^{-1})$ lower bound. Suppose for contradiction that an algorithm $\mathbb{A}$ takes $o(n^2\mathrm{OPT}^{-2}\ln\delta^{-1})$ samples in expectation. Let τ_i be the expected number of samples taken from the i-th arm. For convenience, we define

$$O = \mathrm{OPT} = \sum_{i=1}^{N} \|x_i\|.$$

We construct another sequence of Gaussian distributions $N(u_i, I_d)$ as follows. For each i, we set u_i such that $u_i = x_i + \frac{2O}{N} \cdot \frac{x_i}{\|x_i\|}$. Recall that $v = (0, 0, \ldots, 0)$ is the optimal 1-median solution of $\{x_i\}$. By fixing a coodinate $j \in [d]$, we know that the median value of all $x_{i,j}$ (the j-th entry of x_i) must be 0. By the definition of u_i, it is not hard to see that the median value of all $u_{i,j}$ (the j-th entry of u_i) is 0 for any fixed coordinate $j \in [d]$. Thus, the optimal 1-median center of the instance $\{u_i\}$ is still $v = (0, 0, \ldots, 0)$. We conclude that the optimal 1-median value of $\{u_i\}$ is $\sum_{i=1}^{N} \|u_i\|$. Note that we have

$$\sum_{i=1}^{N} \|u_i\| - \|x_i\| = \sum_{i=1}^{N} \|u_i - x_i\| = \sum_{i=1}^{N} \frac{2O}{N} = 2O.$$

Therefore, we have

$$\sum_{i=1}^{N} \|u_i\| = 3O.$$

Let $\mathcal{C}$ denote the original sequence of arms, and $\mathcal{C}'$ denote the sequence of arms corresponding to u_i. We also let $\mathcal{E}$ be the event that $\mathbb{A}$ outputs a value at most O. Note that if $\mathcal{E}$ happens, $\mathbb{A}$ can not output a 2-approximation solution for $\{u_i\}_{i\in[n]}$.

Clearly, by the assumption of $\mathbb{A}$, we have

$$\Pr_{\mathbb{A},\mathcal{C}}[\mathcal{E}] \geq 1-\delta \text{ and } \Pr_{\mathbb{A},\mathcal{C}'}[\mathcal{E}] \leq \delta.$$

Now we apply Lemma 1.4.3 and Equality (1.2), and we have

$$\sum_{i=1}^{N} \tau_i \cdot \frac{1}{2} \cdot \|x_i - u_i\|^2 \geq d\left(\Pr_{\mathbb{A},\mathcal{C}}[\mathcal{E}], \Pr_{\mathbb{A},\mathcal{C}'}[\mathcal{E}]\right) \geq \Omega(\ln \delta^{-1}).$$

Since

$$\sum_{i=1}^{N} \tau_i \cdot \frac{1}{2} \cdot \|x_i - u_i\|^2 = \sum_{i=1}^{N} \tau_i \cdot \frac{1}{2} \cdot \frac{4O^2}{n^2} = \frac{2O^2}{n^2} \cdot \sum_{i=1}^{N} \tau_i,$$

we have

$$\sum_{i=1}^{N} \tau_i \geq n^2 \cdot \ln \delta^{-1}/2O^2,$$

which completes the proof. ■

By letting $u_i = x_i + \frac{(1+\varepsilon)O}{N} \cdot \frac{x_i}{\|x_i\|}$ in the above proof, we directly have the following corollary.

Corollary 2.4.2. *For any noisy* 1*-median instance in* $\mathbb{R}^d$*, assume that the optimal* 1*-median value is* OPT*. Then any* $(1-\delta)$*-correct algorithm with approximation factor* $1+\varepsilon$ *takes at least* $\Omega(\varepsilon^{-2}n^2\mathrm{OPT}^{-2}\ln\delta^{-1}+n)$ *samples in expectation for any* $\varepsilon > 0$*.*

For the general case, we have the following theorem.

Theorem 2.4.3. *For any noisy* 1*-median instance in* $\mathbb{R}^d$*, assume the optimal* k*-median clustering is* $V = \{v_1, \ldots, v_k\}$ *and the optimal* k*-median value is* OPT*. If for any pair* $v_i, v_j \in V$*, we have* $\|v_i - v_j\| > 6\mathrm{OPT}$*. Then any* $(1-\delta)$*-correct algorithm with approximation factor 2 takes at least* $\Omega(n^2\mathrm{OPT}^{-2}\ln\delta^{-1}+n)$ *samples in expectation.*

Proof. We first show an $\Omega(n)$ lower bound. Suppose for contradiction that a $(1-\delta)$-correct algorithm $\mathbb{A}$ takes $o(n)$ samples in expectation. Now, suppose we pick an arm x_i uniformly at random, and replace x_i with an arm a extremely far from the rest arms. Then clearly the optimal k-median clustering of this modified instance should

include a. But $\mathbb{A}$ can only detect this change with $o(1)$ probability, since it can touch at most $o(n)$ arms. Therefore, $\mathbb{A}$ cannot be $(1-\delta)$-correct with $o(n)$ samples.

The proof of an $\Omega(n^2 \cdot \ln \delta^{-1}/\mathrm{OPT}^2)$ lower bound is similar to Theorem 2.4.1. The optimal solution V partitions $\{x_i\}$ into different clusters according to their distance to V. W.l.o.g., assume $x_1, x_2, \ldots, x_m$ are points satisfying that $v_1 = \arg\min_{v \in V} \|x_i - v\|$ $(1 \le i \le m)$. We construct another sequence of Gaussian distributions $N(u_i, I_d)$ as follows. For each $1 \le i \le m$, we set u_i such that $u_i = x_i + \frac{2\mathrm{OPT}}{n} \cdot \frac{x_i - v_1}{\|x_i - v_1\|}$. Observe that $\|u_i - v_1\| = \|x_i - v_1\| + 2\mathrm{OPT}/n$. For other clusters, we construct u_i respectively.

We first show that V is still the optimal k-median clustering of $\{u_i\}$. Assume V^* is the optimal k-median clustering of $\{u_i\}$. Again, V^* partitions $\{u_i\}$ into different clusters. We claim that each cluster of $\{u_i\}$ according to V^* corresponds to a cluster of $\{x_i\}$ according to V. Note that

$$\sum_{i=1}^{n} \min_{v \in V} \|u_i - v\| \le \sum_{i=1}^{n} \min_{v \in V} \|x_i - v\| + n \cdot 2\mathrm{OPT}/n = 3\mathrm{OPT}. \tag{2.7}$$

W.l.o.g., suppose there exists two points x_1 and x_2 belong to different clusters according to V, while u_1 and u_2 belong to the same cluster v^* according to V^*. W.l.o.g., assume that the closest point in V for x_1 and x_2 are v_1 and v_2 respectively. Then by the triangle inequality and the assumption that $\|v_1 - v_2\| > 6\mathrm{OPT}$, we have

$$\|u_1 - u_2\| > \|v_1 - v_2\| - \|v_1 - x_1\| - \|x_1 - u_1\| - \|v_2 - x_2\| - \|x_2 - u_2\| > 6\mathrm{OPT} - \mathrm{OPT} - 4\mathrm{OPT}/n > 3\mathrm{OPT}.$$

It implies that

$$\sum_{i=1}^{n} \min_{v \in V} \|u_i - v\| \overset{\text{Defn. of } V^*}{\ge} \sum_{i=1}^{n} \min_{v \in V^*} \|u_i - v\| \ge \|u_1 - v^*\| + \|u_2 - v^*\| \overset{\text{triangle ineq.}}{\ge} \|u_1 - u_2\| > 3\mathrm{OPT},$$

which is a contradiction with Inequality (2.7). Thus, x_1 and x_2 belong to different clusters according to V, then u_1 and u_2 must belong to different clusters according to V^*. It implies that each cluster of $\{x_i\}$ according to V corresponds to at least one cluster of $\{u_i\}$ according to V^*. However, we have $|V^*| = |V| = k$. Thus, each cluster

of $\{u_i\}$ according to V^* corresponds to a cluster of $\{x_i\}$ according to V.

Then by the same argument as in Theorem 2.4.1, we can prove that $V^* = V$. Finally, since the partition of $\{u_i\}$ remains the same, we have

$$\sum_{i=1}^{n} \min_{v \in V} \|u_i - v\| = \sum_{i=1}^{n} \min_{v \in V} \|x_i - v\| + n \cdot 2\mathrm{OPT}/n = 3\mathrm{OPT}.$$

By the same argument as in Theorem 2.4.1, we know that any $(1-\delta)$-correct algorithm with approximation factor 2 takes at least $\Omega(n^2 \mathrm{OPT}^{-2} \ln \delta^{-1})$ samples in expectation to distinguish $\{x_i\}$ and $\{u_i\}$. It completes the proof. ■

2.4.2 Worst-Case Lower Bound

Now we prove a worst-case lower bound of the noisy k-median problem. Precisely, we show that there exists a family of noisy k-median instances where $\Omega(n+\sqrt{d}n^2\mathrm{OPT}^{-2})$ samples are required for any 0.9-correct algorithm. This result shows that the factor $\sqrt{d}$ is also necessary in the upper bound. We first need the following lemma for preparation.

Lemma 2.4.4. *There is no 0.9-correct algorithm that takes $o(\sqrt{d}/\varepsilon^2)$ samples in expectation from an d-dimensional Gaussian $N(\mu, I_d)$ distinguishes between the cases*

1. $\mu = 0$
2. $\|\mu\| > \varepsilon$.

Proof. Note that this lemma is very similar to Theorem C.2 in [36]. [3] The only difference is that we consider the number of samples in expectation.

Suppose for sake of contradiction that such an algorithm exists. If the probability that it distinguishes the two cases with $o(\sqrt{d}/\epsilon^2)$ samples is larger than $1/3$. Then by Theorem C.2 in [36], we have that the accuracy of this algorithm is less than $1 - 1/3 \times 1/3 = 8/9$, which is a contradiction. Otherwise, the algorithm takes

[3] The proof of Theorem C.2 in [36] has a mistake. The last equality should be $\|\Sigma - I\|_F^2 = nk^2(2\epsilon^2/n)^2 = 4k^2\epsilon^4/n = o(1)$. However, it does not affect the correctness of the theorem.

$\Omega(\sqrt{d}/\epsilon^2)$ samples with probability at least 2/3. Then the algorithm takes $\Omega(\sqrt{d}/\epsilon^2)$ samples in expectation, which is also a contradiction. ■

We are ready to prove the following theorem.

Theorem 2.4.5. *There exists an infinite sequence of noisy k-median instances, such that any 0.9-correct algorithm for any instance with approximation factor 2 takes at least*

$$\Omega(\sqrt{d}n^2\mathrm{OPT}^{-2} + n)$$

samples in expectation. Here, OPT *is the optimal k-median value.*

Proof. Consider the following sequence of $(2n+2)$-arms $\mathcal{C} = (a_0 = N(x_0, I_d), b_0 = N(y_0, I_d), a_1 = N(x_1, I_d), b_1 = N(x_1, I_d), \ldots, a_n = N(x_n, I_d), b_n = (x_n, I_d))$ satisfying the following property:

1. $\|y_0 - x_0\| = \sqrt{n}$.
2. The distance between any x_i and x_j $(0 \le i < j \le n)$ is far away enough, say 2^{n^2}.

Let $k = n + 1$. Assume that $\mathbb{A}$ is a 0.9-correct algorithm with approximation factor 2. By the same argument as in Lemma 2.4.3, $\mathbb{A}$ must take $\Omega(n)$ samples in expectation. Then suppose for sake of contradiction that $\mathbb{A}$ takes $o(\sqrt{d}n^2\mathrm{OPT}^{-2})$ samples in expectation. Let $V = \{x_0, x_1, \ldots, x_n\}$. Observe that the optimal k-median value of $\mathcal{C}$ is exactly $\mathrm{OPT} = \sum_{i=0}^{n} \min_{v \in V} \|y_i - v\| = \sqrt{n}$. Thus, $\mathbb{A}(\mathcal{C})$ outputs a value at most $2\sqrt{n}$ with probability at least 0.9.

Construct another sequence of $(2n+2)$ independent arms $\mathcal{C}' = (a'_0 = N(x_0, I_d), b'_0 = N(y_0, I_d), a'_1 = N(x_1, I_d), b'_1 = N(x_1 + \xi, I_d), \ldots, a'_n = N(x_n, I_d), b'_n = (x_n + \xi, I_d))$. Here, $\xi \in \mathbb{R}^d$ satisfies that $1/\sqrt{n} < \|\xi\| \le 2nd$. In this case, the optimal k-median value of $\mathcal{C}'$ is exactly $\mathrm{OPT}' = \sum_{i=0}^{n} \min_{v \in V} \|y_i - v\| > 2\sqrt{n}$. Thus, $\mathbb{A}(\mathcal{C}')$ outputs a value larger than $2\sqrt{n}$ with probability at least 0.9.

Now we construct an algorithm $\hat{\mathbb{A}}$ to distinguish the two cases in Lemma 2.4.4: $\mu = 0$ or $\|\mu\| > 1/\sqrt{n}$. Firstly, $\hat{\mathbb{A}}$ take 1 sample x. If $\|x\| > nd$, output $\|\mu\| > \epsilon$.

Otherwise, $\hat{\mathbb{A}}$ construct a $(2n+2)$-arm instance as follows: $\hat{\mathcal{C}} = (\hat{a}_0 = N(x_0, I_d), \hat{b}_0 = N(y_0, I_d), \hat{a}_1 = N(x_1, I_d), \hat{b}_1 = N(x_1 + \mu, I_d), \ldots, \hat{a}_n = N(x_n, I_d), \hat{b}_n = (x_n + \mu, I_d))$. Let $V = \{x_0, x_1, \ldots, x_n\}$. We obtain a noisy $(n+1)$-median instance.

Run algorithm $\mathbb{A}$ on this instance. This can be done by the following simulation. If $\mathbb{A}$ wants to take a sample from arm $\hat{a}_i$ $(0 \leq i \leq n)$ or arm $\hat{b}_0$, $\hat{\mathbb{A}}$ directly draws a sample from the corresponding Gaussian distribution. If $\mathbb{A}$ wants to take a sample from arm $\hat{b}_i$ $(1 \leq i \leq n)$, $\hat{\mathbb{A}}$ draws a sample y from the unknown distribution $N(\mu, I_d)$, and produces $y + x_i$ as a sample drawn from arm $\hat{b}_i$. If $\mathbb{A}(\hat{\mathcal{C}})$ outputs a value at most $2\sqrt{n}$, then $\hat{\mathbb{A}}$ outputs $\|\mu\| = 0$. Otherwise if $\mathbb{A}(\hat{\mathcal{C}})$ outputs a value larger than $2\sqrt{n}$, then $\hat{\mathbb{A}}$ outputs $\|\mu\| > 1/\sqrt{n}$. Note that $\hat{\mathbb{A}}$ only takes $o(n^2\sqrt{d}/\sqrt{n}^2) = o(n\sqrt{d})$ samples in expectation.

It remains to prove that the correctness of $\hat{\mathbb{A}}$ is at least 0.9. Then by Lemma 2.4.4, $\hat{\mathbb{A}}$ must take $\Omega(n\sqrt{d})$ in expectation which leads to a contradiction and finishes the proof. Firstly, if $\|\mu\| > 2nd$, then with probability 0.99, $\mathbb{A}$ take a sample x with $\|x\| > nd$ and successfully distinguish the two cases. Otherwise if $1/\sqrt{n} < \|\mu\| \leq 2nd$, $\mathbb{A}(\hat{\mathcal{C}})$ will output a value larger than $2\sqrt{n}$ with probability at least 0.9 by the assumption of $\mathbb{A}$. Finally, if $\mu = 0$, $\mathbb{A}(\hat{\mathcal{C}})$ will output a value at most $2\sqrt{n}$ with probability at least 0.9. Thus, $\hat{\mathbb{A}}$ can distinguish the two cases with success probability at least 0.9, which is a contraction. ■

Chapter 3

Robust Coreset and Property Testing

In this Chapter, we consider robust coresets for the (k, z)-clustering problem with outliers (see Definition 1.2.2). We generalize and improve the prior result [39] for Euclidean space, and prove the existence of robust coresets with smaller size in doubling metrics. The following is the main theorem of this Chapter.

Theorem 3.0.6. *Let $M(X, d)$ be a doubling metric space (a d-dimensional Euclidean space resp.). Suppose S is a uniform independent sample of Γ (Γ' resp.) points from X, where*

$$\Gamma = O\left(\frac{k}{\alpha^2}(\mathrm{ddim}(M) \cdot \log(z/\varepsilon) + \log k + \log\log(1/\tau)) + \frac{\log(1/\tau)}{\alpha^2}\right)$$

and

$$\Gamma' = O\left(\frac{1}{\alpha^2}(kd \log k + \log(1/\tau))\right).$$

Then with probability at least $1-\tau$, $\mathcal{S}$ is an (α, ε)-robust coreset ($(\alpha, 0)$-robust coreset resp.) for the (k, z)-clustering problem with outliers.

3.1 Approximation to Robust Coreset

Our main idea is to construct an ε-approximation and show that an ε-approximation for the range space already induces a robust coreset. We consider the functional representation of the problem as follows:

Definition 3.1.1 (Robust Coreset for a Set of Functions)**.** Assume $0 < \alpha, \varepsilon < \frac{1}{4}$. Let $\mathcal{G}$ be a finite set of functions $[X]^k \to \mathbb{R}_{\geq 0}$. For any $0 < \gamma < 1$, $C \in [X]^k$ and $\mathcal{S} \subseteq \mathcal{G}$, let

$$\mathcal{S}^{-\gamma}(C) := \min_{\mathcal{S}' \subseteq \mathcal{S}: |\mathcal{S}'| = \lceil (1-\gamma)|\mathcal{S}| \rceil} \sum_{g \in \mathcal{S}'} g(C),$$

which is the sum of the smallest $\lceil (1-\gamma)|\mathcal{S}| \rceil$ values $g(C)$. Then a subset $\mathcal{S} \subseteq \mathcal{G}$ is called an (α, ε)-robust coreset of $\mathcal{G}$ if for any $\alpha < \gamma < 1 - \alpha$ and $C \in [X]^k$,

$$(1-\varepsilon) \cdot \frac{\mathcal{G}^{-(\gamma+\alpha)}(C)}{|\mathcal{G}|} \leq \frac{\mathcal{S}^{-\gamma}(C)}{|\mathcal{S}|} \leq (1+\varepsilon) \cdot \frac{\mathcal{G}^{-(\gamma-\alpha)}(C)}{|\mathcal{G}|}. \tag{3.1}$$

Remark 3.1.2. To reduce the problem of constructing a robust coreset for clustering to the problem for functions, for $x \in X$, let $g_x(\cdot)$ be a function from $[X]^k$ to $\mathbb{R}_{\geq 0}$ such that $g_x(C) = d^z(x, C)$. Let $\mathcal{G} := \{g_x \mid x \in X\}$.

We note that our definition is slightly different from that in [39, Definition 8.1] [1]. In particular, in Euclidean spaces, one can check that an $(\varepsilon\gamma/4, 0)$-robust coreset is a (γ, ε)-coreset in [39, Definition 8.1].

Next, we prove the following simple connection between α-approximation of $(\mathcal{G}, \text{ranges}(\mathcal{G}))$ and robust coreset of $\mathcal{G}$ in Lemma 3.1.4. This lemma improves [39, Theorem 8.3] in which they show that an $(\varepsilon^2\gamma/63)$-approximation is a (γ, ε)-coreset[2]. First we need the following simple formulas.

Claim 3.1.3. *For any $\gamma \in (\alpha, 1-\alpha)$ and $C \in [X]^k$, the following equations hold:*

[1] In fact, our definition is more general. It is unclear whether their result applies to our definition.

[2] Consider the (γ, ε)-coreset in [39, Definition 8.1]. Since an $(\varepsilon\gamma/4, 0)$-robust coreset is a (γ, ε)-coreset, our Lemma 3.1.4 implies that an $(\varepsilon\gamma/8)$-approximation is a (γ, ε)-coreset.

$$\frac{\mathcal{S}^{-\gamma}(C)}{|\mathcal{S}|} = \int_0^\infty \left(\frac{\lceil (1-\gamma)|\mathcal{S}| \rceil}{|\mathcal{S}|} - \frac{|\mathcal{S} \cap \text{range}(\mathcal{G}, C, r)|}{|\mathcal{S}|} \right)_+ dr, \tag{3.2}$$

$$\frac{\mathcal{G}^{-(\gamma+\alpha)}(C)}{|\mathcal{G}|} = \int_0^\infty \left(\frac{\lceil (1-\gamma-\alpha)|\mathcal{G}| \rceil}{|\mathcal{G}|} - \frac{|\text{range}(\mathcal{G}, C, r)|}{|\mathcal{G}|} \right)_+ dr, \tag{3.3}$$

$$\frac{\mathcal{G}^{-(\gamma-\alpha)}(C)}{|\mathcal{G}|} = \int_0^\infty \left(\frac{\lceil (1-\gamma+\alpha)|\mathcal{G}| \rceil}{|\mathcal{G}|} - \frac{|\text{range}(\mathcal{G}, C, r)|}{|\mathcal{G}|} \right)_+ dr. \tag{3.4}$$

Proof. We only proof the first one. The other two Equations (3.3) and (3.4) can be proved in the same manner. Let $\mathcal{D}$ be the collection of functions $g \in \mathcal{S}$ with the smallest $\lceil (1-\gamma)|\mathcal{S}| \rceil$ values $g(C)$. Using integration, we know that

$$\frac{\mathcal{S}^{-\gamma}(C)}{|\mathcal{S}|} = \int_0^\infty \frac{\left| \{ g(C) > r \mid g \in \mathcal{D} \} \right|}{|\mathcal{S}|} dr.$$

By definition, we have

$$\begin{aligned} \frac{\left| \{ g(C) > r \mid g \in \mathcal{D} \} \right|}{|\mathcal{S}|} &= \frac{|\mathcal{D} \setminus \text{range}(\mathcal{G}, C, r)|}{|\mathcal{S}|} = \frac{\left(|\mathcal{D}| - |\mathcal{S} \cap \text{range}(\mathcal{G}, C, r)| \right)_+}{|\mathcal{S}|} \\ &= \left(\frac{\lceil (1-\gamma)|\mathcal{S}| \rceil}{|\mathcal{S}|} - \frac{|\mathcal{S} \cap \text{range}(\mathcal{G}, C, r)|}{|\mathcal{S}|} \right)_+, \end{aligned}$$

which proves Equation (3.2). ■

Lemma 3.1.4. *If $\mathcal{S}$ is an $\frac{\alpha}{2}$-approximation of $(\mathcal{G}, \text{ranges}(\mathcal{G}))$ such that $|\mathcal{S}|, |\mathcal{G}| \geq 2/\alpha$, then $\mathcal{S}$ is an $(\alpha, 0)$-robust coreset of $\mathcal{G}$.*

Proof. Let $\mathcal{S} \subseteq \mathcal{G}$ be an α-approximation of $(\mathcal{G}, \text{ranges}(\mathcal{G}))$. We prove $\mathcal{S}$ is also an $(\alpha, 0)$-robust coreset of $\mathcal{G}$. Since $\mathcal{S}$ is an $\frac{\alpha}{2}$-approximation of $\mathcal{G}$, for any $C \in [X]^k$ and $r \geq 0$,

$$\left| \frac{|\text{range}(\mathcal{G}, C, r)|}{|\mathcal{G}|} - \frac{|\mathcal{S} \cap \text{range}(\mathcal{G}, C, r)|}{|\mathcal{S}|} \right| \leq \frac{\alpha}{2}. \tag{3.5}$$

So we have that

$$\begin{aligned}\left(\frac{\lceil(1-\gamma-\alpha)|\mathcal{G}|\rceil}{|\mathcal{G}|}-\frac{|\text{range}(\mathcal{G},C,r)|}{|\mathcal{G}|}\right)_+ &\leq \left(\frac{\lceil(1-\gamma-\alpha)|\mathcal{G}|\rceil}{|\mathcal{G}|}+\frac{\alpha}{2}-\frac{|\mathcal{S}\cap\text{range}(\mathcal{G},C,r)|}{|\mathcal{S}|}\right)_+\\ &\leq \left(\frac{(1-\gamma-\alpha)|\mathcal{G}|+1}{|\mathcal{G}|}+\frac{\alpha}{2}-\frac{|\mathcal{S}\cap\text{range}(\mathcal{G},C,r)|}{|\mathcal{S}|}\right)_+\\ &= \left(\frac{(1-\gamma)|\mathcal{S}|}{|\mathcal{S}|}-\frac{\alpha}{2}+\frac{1}{|\mathcal{G}|}-\frac{|\mathcal{S}\cap\text{range}(\mathcal{G},C,r)|}{|\mathcal{S}|}\right)_+\\ &\leq \left(\frac{\lceil(1-\gamma)|\mathcal{S}|\rceil}{|\mathcal{S}|}-\frac{|\mathcal{S}\cap\text{range}(\mathcal{G},C,r)|}{|\mathcal{S}|}\right)_+\end{aligned}$$

The first inequality holds due to Inequality (3.5) and the last follows because $|\mathcal{G}| \geq 2/\alpha$.

Together with (3.2) and (3.3), we have that

$$\frac{\mathcal{G}^{-(\gamma+\alpha)}(C)}{|\mathcal{G}|} \leq \frac{\mathcal{S}^{-\gamma}(C)}{|\mathcal{S}|}.$$

Similarly, by (3.2), (3.4) and (3.5), we can also show that

$$\frac{\mathcal{S}^{-\gamma}(C)}{|\mathcal{S}|} \leq \frac{\mathcal{G}^{-(\gamma-\alpha)}(C)}{|\mathcal{G}|},$$

which completes the proof. ■

In the d-dimensional Euclidean space, one can utilize a $(\gamma\varepsilon/8)$-approximation to construct a (γ,ε)-coreset of [39, Definition 8.1]. Using the improved Lemma 3.1.4, we can improve the robust coreset size in [39, Definition 8.1] from $O(kd\log k\cdot\gamma^{-2}\varepsilon^{-4})$ [3] to $O(kd\log k\cdot\gamma^{-2}\varepsilon^{-2})$.

Proof. (proof of Theorem 3.0.6) For the Euclidean space $\mathbb{R}^d$, by [69], we can construct an $\frac{\alpha}{2}$-approximation of $\mathcal{G}$ defined as in Remark 3.1.2, by taking $O(\frac{kd\log k}{\alpha^2})$ uniform samples from X. Then by Lemma 3.1.4, we complete the proof for the Euclidean space.

Since dealing with doubling metrics requires more involved techniques which are

[3] The size stated in [39] is $O(kd\gamma^{-2}\varepsilon^{-4})$. We defer interesting readers to [9, Section 5] to see why an additional $\log k$ factor is required.

less relevant with this paper. We omit all the details here but refer the interested readers to our paper [58, Section 6.2.1]. ■

3.2 Application to Property Testing

In this section, we show some applications of robust coreset to property testing. We start with the following definition that captures the notion of bi-criteria algorithms.

Definition 3.2.1. Let $M(X,d)$ be a metric space. Let $\lambda \geq 1$, $0 < \alpha < 1/4$ and $\alpha < \gamma < 1-\alpha$. We say A is a (λ,γ,α)-approximation algorithm for the (k,z)-clustering problem with outliers, if A returns a number Λ such that $\min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma+\alpha)}(X,C) \leq \Lambda \leq \lambda \cdot \min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma-\alpha)}(X,C)$.

Theorem 3.2.2 (Testing of (k,z)-clustering)**.** *Let $M(X,d)$ be a doubling metric space (d-dimensional Euclidean space resp.). Let $\lambda \geq 1$, $0 < \alpha < 1/4$ and $\alpha < \gamma < 1-\alpha$. Suppose there is a (λ,γ,α)-approximation algorithm for the (k,z)-clustering problem with outliers, which runs in time $T(|X|,\lambda,\gamma,\alpha)$. Then for any $\Delta > 0$ and $0 < \varepsilon < 1/4$, there is an algorithm satisfying*

1. *if $\min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma-\alpha)}(X,C) \leq \Delta$, it accepts with probability $1-\tau$;*
2. *if $\min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma+\alpha)}(X,C) \geq \lambda(1+\varepsilon)\cdot\Delta$, it rejects with probability $1-\tau$,*

with running time $T(\Gamma,\gamma,\lambda,\frac{\alpha}{2}) + \Gamma^2$, where

$$\Gamma := O\left(\frac{k}{\alpha^2}(\mathrm{ddim}(M)\cdot\log(z/\varepsilon) + \log k + \log\log(1/\tau)) + \frac{\log(1/\tau)}{\alpha^2}\right)$$

for doubling metrics and

$$\Gamma := O\left(\frac{1}{\alpha^2}(kd\log k + \log(1/\tau))\right)$$

for d-dimensional Euclidean space.

Proof. Consider the following algorithm:

1. Take a uniformly independent sample S of size Γ from X.

2. Run the $(\lambda, \gamma, \frac{\alpha}{2})$-approximation algorithm on S. Suppose the output is Γ.

3. Accept if $\Gamma \leq \frac{(1+\varepsilon/4)\lambda\Gamma}{|X|} \cdot \Delta$, and reject otherwise.

By Theorem 3.0.6, with probability at least $1-\tau$, S is an $(\frac{\alpha}{2}, \frac{\varepsilon}{4})$-robust coreset for X [4]. In the following, we condition on the event that S is an $(\frac{\alpha}{2}, \frac{\varepsilon}{4})$-robust coreset for X. Hence, for any $C \in [X]^k$ and $\alpha < \gamma < 1-\alpha$, we have

$$(1-\varepsilon/4) \cdot \frac{\mathcal{K}_z^{-(\gamma+\frac{\alpha}{2})}(X,C)}{|X|} \leq \frac{\mathcal{K}_z^{-\gamma}(S,C)}{|S|} \leq (1+\varepsilon/4) \cdot \frac{\mathcal{K}_z^{-(\gamma-\frac{\alpha}{2})}(X,C)}{|X|}. \tag{3.6}$$

Recall that Λ is the output of the $(\lambda, \frac{\alpha}{2})$-approximation algorithm. Then by Definition 3.2.1 and Inequality (3.6), we have

$$\Lambda < \lambda \cdot \min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma-\frac{\alpha}{2})}(S,C) \leq \frac{(1+\varepsilon/4)\lambda\Gamma}{|X|} \cdot \min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma-\alpha)}(X,C), \tag{3.7}$$

and

$$\Lambda \geq \min_{C\in[X]^k} \mathcal{K}_z^{-\gamma}(S,C) \geq \frac{(1-\varepsilon/4)\Gamma}{|X|} \cdot \min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma+\alpha)}(X,C) \tag{3.8}$$

If $\min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma-\alpha)}(X,C) \leq \Delta$, we have $\Gamma \leq \frac{(1+\varepsilon/4)\lambda\Gamma}{|X|} \cdot \Delta$ by Inequality (3.7). In this case, our algorithm accepts. On the other hand, if $\min_{C\in[X]^k} \mathcal{K}_z^{-(\gamma+\alpha)}(X,C) \geq \lambda(1+\varepsilon) \cdot \Delta$, we have

$$\Gamma \overset{\text{Ineq. (3.8)}}{\geq} \frac{(1-\varepsilon/4)\Gamma}{|X|} \cdot \lambda(1+\varepsilon) \cdot \Delta > \frac{(1+\varepsilon/4)\lambda\Gamma}{|X|} \cdot \Delta.$$

In this case, our algorithm rejects. It completes the proof. ■

Remark 3.2.3. The $(\lambda, \gamma, \alpha)$-approximation algorithm for the (k, z)-clustering problem with outliers is used as a subroutine in our testing algorithm. If we use exhaustive search, we obtain a $(1, \gamma, 0)$-approximation algorithm with running time exponential

[4] Recall that in Euclidean space, S is actually an $(\frac{\alpha}{2}, 0)$-robust coreset, but the weaker property is sufficient here.

in $|S|$ for (k, z)-clustering with outliers. If we use the approximation algorithm by Charikar et al. [27], we have a $(4(1 + \lambda^{-1}), \gamma, \lambda\gamma)$-approximation algorithm with polynomial running time for the $(k, 1)$-clustering problem with outliers.

Appendix A

Missing Proofs

Proof of Lemma 2.1.2. Let $v_j^* = \operatorname{argmin}_{v \in V} \|x_j - v\|$ where ties are broken arbitrarily. Suppose $r_i \leq \operatorname{dist}(x_j, V)/12$. Observe that at the i-th round, we have

$$d_{jv} = \min_{y \in B(x_j^{(i)}, 3r_i)} \|y - v\| = \max\{0, \|x_j^{(i)} - v\| - 3r_i\} \geq \|x_j^{(i)} - v\| - 3r_i$$
$$\overset{\text{triangle ineq.}}{\geq} \|x_j - v\| - \|x_j^{(i)} - x_j\| - 3r_i \overset{r_i \leq \operatorname{dist}(x_j, V)/12 \text{ and } \mathcal{E}}{\geq} 12r_i - r_i - 3r_i = 8r_i > 0.$$

Also note that

$$\begin{aligned} c_{jv_j^*} = & \max_{y \in B(x_j^{(i)}, 3r_i)} \|y - v_j^*\| = \|x_j^{(i)} - v_j^*\| + 3r_i \\ \leq & \|x_j - v_j^*\| + \|x_j - x_j^{(i)}|_2 + 3r_i && \text{(triangle ineq.)} \\ \leq & \|x_j - v_j^*\| + 4r_i && (\mathcal{E}) \\ = & \operatorname{dist}(x_j, V) + 4r_i, \end{aligned}$$

and for every $v \neq v_j^*$

$$\begin{aligned} d_{jv} &= \min_{y \in B(x_j^{(i)}, 3r_i)} \|y - v\| = \max\{0, \|x_j^{(i)} - v\| - 3r_i\} \\ &\geq \|x_j - v\| - \|x_j - x_j^{(i)}\| - 3r_i \\ &\geq \operatorname{dist}(x_j, V) - 4r_i. \end{aligned}$$

Hence we have

$$\frac{c_{jv_j^*}}{d_{jv}} \leq \frac{\text{dist}(x_j, V) + 4r_i}{\text{dist}(x_j, V) - 4r_i} \leq \frac{12r_i + 4r_i}{12r_i - 4r_i} = 2,$$

which implies that the "IF" sentence in Line 9 is satisfied. Hence flag(j) has been set to be "TRUE" at the i-th round.

On the other hand, if $r_i > \text{dist}(x_j, V)/2$, we have that

$$\|x_j^{(i)} - v_j^*\| \overset{\text{triangle ineq.}}{\leq} \|x_j - x_j^{(i)}\| + \|x_j - v_j^*\| \overset{\mathcal{E}}{\leq} r_i + \text{dist}(x_j, V) \overset{r_i > \text{dist}(x_j,V)/2}{<} r_i + 2r_i = 3r_i.$$

It means that $v_j^* \in B(x_j^{(i)}, 3r_i)$ which implies that $d_{jv_j^*} = 0$. Hence Line 9 is not satisfied which implies that flag(j) remains "FALSE" at this round.

Consequently, if flag(j) is set to be "TRUE" at the i-th round of the while-loop, we have $r_i \leq \text{dist}(x_j, V)/2$. Hence

$$\text{dist}(x_j^{(i)}, V) \overset{\text{triangle ineq.}}{\leq} \text{dist}(x_j, V) + \|x_j^{(i)} - x_j\| \overset{\mathcal{E}}{\leq} \text{dist}(x_j, V) + r_i \overset{r_i \leq \text{dist}(x_j,V)/2}{\leq} \frac{3}{2}\text{dist}(x_j, V)$$

and

$$\text{dist}(x_j^{(i)}, V) \overset{\text{triangle ineq.}}{\geq} \text{dist}(x_j, V) - \|x_j^{(i)} - x_j\| \overset{\mathcal{E}}{\geq} \text{dist}(x_j, V) - r_i \overset{r_i \leq \text{dist}(x_j,V)/2}{\geq} \frac{1}{2}\text{dist}(x_j, V),$$

which complete the proof. ■

Proof of Lemma 2.1.4. Suppose $\mathcal{E}$ happens and $r_i < \frac{\text{OPT}}{200n}$ in Line 3 of the i-th round. Then $r_{i-1} < 0.01\text{OPT}/n$ and $r_{i-2} < 0.02\text{OPT}/n$.

Let OPT_1 denote the optimal k-median value on $X_{i-1} = \{x_j^{(i-1)} : j = 1, 2, \ldots, n\}$. Since A_{i-1} is the output of $\mathsf{BPRS}(N, k, d, X_{i-1})$, by optimality, we know that,

$$C_2^{(i-1)} = \text{cost}(X_{i-1}, A_{i-1}) \geq \text{OPT}_1.$$

By the triangle inequality, we have

$$\mathrm{OPT}_1 \geq \mathrm{OPT} - \sum_{j=1}^{n} \|x_j - x_j^{(i-1)}\| \overset{\mathcal{E}}{\geq} \mathrm{OPT} - n \cdot r_{i-1} \overset{r_{i-1} < \frac{\mathrm{OPT}}{100n}}{\geq} \mathrm{OPT} - \frac{\mathrm{OPT}}{100} = 0.99\mathrm{OPT}.$$

Let OPT_2 be the optimal k-median value on $X_{i-2} = \{x_j^{(i-2)} : j = 1, 2, \ldots, n\}$. Since A_{i-2} is the output of $\mathsf{BPRS}(N, k, d, X_{i-2})$, we know that,

$$C_2^{(i-2)} = \mathrm{cost}(X_{i-2}, A_{i-2}) \leq \alpha \mathrm{OPT}_2 < 6\mathrm{OPT}_2.$$

By the triangle inequality, we have

$$\mathrm{OPT}_2 \leq \mathrm{OPT} + \sum_{j=1}^{n} \|x_j - x_j^{(i-2)}\| \overset{\mathcal{E}}{\leq} \mathrm{OPT} + n \cdot r_{i-2} \overset{r_{i-2} < \frac{\mathrm{OPT}}{50n}}{\leq} \mathrm{OPT} + 0.02\mathrm{OPT} = 1.02\mathrm{OPT}.$$

Hence $C_2^{(i-1)} \geq 0.99\mathrm{OPT}$ and $C_2^{(i-2)} \leq 6 \times 1.02\mathrm{OPT} = 6.18\mathrm{OPT}$.

Now consider $C_1^{(i-1)}$. Conditioned on $\mathcal{E}$, we have

$$\begin{aligned}
C_1^{(i-1)} =& \mathsf{DisNATKP}(\{\mathcal{O}(x_{i-1}) : i = 1, 2, \ldots, N\}, A_{i-2}, \frac{\delta}{100(i-1)^2}) \\
\leq& \frac{3}{2}\mathrm{cost}(X, A_{i-2}) && \text{(Lemma 2.1.1)} \\
\leq& \frac{3}{2}(\mathrm{cost}(X_{i-2}, A_{i-2}) + \sum_{j=1}^{n} \|x_j - x_j^{(i-2)}\|) && \text{(triangle ineq.)} \\
\leq& \frac{3}{2}(C_2^{(i-2)} + n r_{i-2}) && (\mathcal{E}) \\
\leq& 9.18\mathrm{OPT} + 0.03\mathrm{OPT} = 9.21\mathrm{OPT}. && (r_{i-2} < \frac{\mathrm{OPT}}{50n})
\end{aligned}$$

Hence $C_1^{(i-1)}/C_2^{(i-1)} \leq \frac{9.21}{0.99} < 10$. It means that the algorithm terminates at the $(i-1)$-th round or before which is a contradiction with the assumption that the last round is i. Therefore, we have $r_i \geq \frac{\mathrm{OPT}}{200n}$ in Line 3 of the i-th round.

In Line 4, observe that r_i is set to be $\frac{C_1^{(i)}}{80n}$ if $r_i \geq \frac{C_1^{(i)}}{80n}$. However, it does not make $r_i < \frac{\mathrm{OPT}}{200n}$ since $C_1^{(i)} \overset{\text{Lemma 2.1.1}}{\geq} \frac{2}{5}\mathrm{cost}(X, A_{i-1}) \geq \frac{2}{5}\mathrm{OPT}$. Overall, $r_i \geq \frac{\mathrm{OPT}}{200n}$. ■

Proof of Lemma 2.2.4. For every $j \in [n]$, let β_{ij} denote the indicator function of

$j \in R_i$. Fix a $j \in [n]$, let i^* be the smallest number such that $\beta_{ij} = 1$. If such i^* exists, by the definition of R_i and the fact that $C_{i+1} = C_i/2$, we have

$$|x_j - x| \geq C_{i^*}/2 > \frac{1}{2}(C_{i^*} + C_{i^*+1} + \ldots + C_L) = \frac{1}{2}\sum_{i=1}^{L} \beta_{ij} C_i.$$

Otherwise if such i^* does not exist, we have $\beta_{1j} = \ldots = \beta_{Lj} = 0$. Hence we have

$$|x_j - x| \geq 0 = \frac{1}{2}\sum_{i=1}^{L} \beta_{ij} C_i.$$

Thus, we always have $|x_j - x| \geq \frac{1}{2}\sum_{i=1}^{L} \beta_{ij} C_i$. Moreover, $|R_i| = \beta_{i1} + \beta_{i2} + \ldots + \beta_{iL}$. So we have,

$$\mathrm{cost}(X, x) = \sum_{i=1}^{n} |x_j - x| \geq \sum_{j=1}^{n} \frac{1}{2}\left(\sum_{i=1}^{L} C_i \beta_{ij}\right) = \frac{1}{2}\sum_{i=1}^{L} C_i |R_i|.$$

■

Proof of Lemma 2.3.1. We first bound the sample complexity of our algorithm. Recall that $r_i = \min\{1, C_i\}$ and $C_i \leq C_1 = O(\max\{1, C\})$.

At the i-th round, if $C_i \leq C$, we take

$$m_i |S_i| = O(r_i^{-2}(d + \log(nL\delta^{-1})) \cdot n^2 L^2 r_i^{-2} C^{-2} \log(nL\delta^{-1})) = \tilde{O}(dn^2 C^{-2} \log^2 \delta^{-1})$$

many samples. If $C_i > C$, we take

$$nm_i = O(nr_i^{-2}(d + \log(nL\delta^{-1}))) = \tilde{O}(d(n + nC^{-2}) \log \delta^{-1})$$

many samples.

Note that we have $L = \tilde{O}_{n,C}(1)$ many rounds in total. So the sample complexity is upper bounded by

$$\tilde{O}(d(n + n^2 C^{-2}) \log^2 \delta^{-1}),$$

which proves our third claim.

Then we prove our first claim, i.e., if $\text{cost}(X, V) > 10C$ then the algorithm accepts with probability at least $1 - \delta$. We first define two events $\mathcal{E}$ and $\mathcal{E}'$ as follows.

Define event $\mathcal{E}_{ij} = \{\omega : \left\|x_j^{(i)} - x_j\right\| \leq r_i/2\}$ for $i \in [L]$ and $j \in [n]$. By the same proof as in Lemma 2.1.1, we have

$$\Pr[\mathcal{E}_{ij}] \geq 1 - \frac{\delta}{2nL}.$$

Let $\mathcal{E} = \cap_{i\in[L],j\in[n]}\mathcal{E}_{ij}$. Then by the union bound, we have $\Pr[\mathcal{E}] \geq 1 - nL \cdot \frac{\delta}{2nL} = 1 - \delta/2$.

Let $M_i = \{j \in [n] : \text{dist}(x_j, V) > 2C_i\}$ and let $P_i = S_i \cap M_i$. For every i satisfying that $C_i \leq C$, we define event

$$\mathcal{E}_i' = \{\omega : \left|\frac{n}{|S_i|}|P_i| - |M_i|\right| \leq CL^{-1}C_i^{-1}/8\},$$

and $\mathcal{E}' = \cap_{i:C_i\leq C}\mathcal{E}_i'$.

Since $|S_i| = O(n^2L^2C_i^2C^{-2}\log(\delta^{-1}L))$, by Theorem 1.4.2, we have that

$$\Pr[\mathcal{E}_i'] = 1 - \Pr\left(\left|\frac{n}{|S_i|}|P_i| - |M_i|\right| > CL^{-1}C_i^{-1}/8\right) \geq 1 - \frac{\delta}{2L}.$$

Hence $Pr[\mathcal{E}'] \geq 1 - L \cdot \frac{\delta}{2L} = 1 - \delta/2$. Combining with the conclusion that $\Pr[\mathcal{E}] \geq 1 - \delta/2$, we have $\Pr[\mathcal{E} \cap \mathcal{E}'] \geq 1 - \delta$ by the union bound.

We only need to prove that $T > C$ conditioned on $\mathcal{E} \cap \mathcal{E}'$ since it implies that the algorithm accepts with probability at least $1 - \delta$. If $C_i \leq C$ and $j \in P_i$, then we have

$$\text{dist}(x_j^{(i)}, V) \overset{\text{triangle ineq.}}{\geq} \text{dist}(x_j, V) - \|x_j^{(i)} - x_j\| \overset{j\in P_i \text{ and } \mathcal{E}}{\geq} 2C_i - r_i/2 \overset{r_i\leq C_i}{\geq} 2C_i - C_i/2 \geq C_i.$$

Therefore, each such j must be counted in Line 9 which implies that $N_i \geq \frac{n}{|S_i|}|P_i|$. Moreover,

$$N_i \geq \frac{n}{|S_i|}|P_i| \overset{\mathcal{E}'}{\geq} |M_i| - CL^{-1}C_i^{-1}/8.$$

On the other hand, if $C > C_i$, since the algorithm checks every $x_j, j \in [n]$, we know that if $j \in M_i$ then j contributes one to N_i. It implies that $N_i \geq |M_i|$. Therefore, we

conclude that for every i,

$$N_i \geq |M_i| - CL^{-1}C_i^{-1}/8. \tag{A.1}$$

Next, we consider the following two cases.

1) If there is some j such that $\mathrm{dist}(x_j, V) > 3C$ then for some $i \in [L]$ satisfying that $C < C_i \leq 2C$,

$$\mathrm{dist}(x_j^{(i)}, V) \overset{\text{triangle ineq.}}{\geq} \mathrm{dist}(x_j, V) - \|x_j^{(i)} - x_j\| \overset{\mathcal{E}}{\geq} 3C - r_i/2 \overset{r_i \leq C_i \leq 2C}{>} 3C - 2C = C.$$

Hence at the i-th round, we have $N_i \geq 1$ which implies $T \geq C_i > C$.

2) Assume that $\mathrm{dist}(x_j, V) \leq 3C$ for every $j \in [n]$. In this case, we claim that

$$\mathrm{cost}(X, V) \leq 2\sum_{i=1}^{L} C_i|M_i| + \frac{C}{10}. \tag{A.2}$$

For every $j \in [n]$, let α_{ij} denote the indicator function of $j \in M_i$. Since $\mathrm{cost}(x_j, V) \leq 3C < 2C_1$, we have $\alpha_{1j} = 0$. Hence there must exist some $i^* \in \{1, 2, \ldots, L\}$ which is the smallest number such that $\alpha_{ij} = 0$. Then by the definition of M_i and $C_{i+1} = C_i/2$, we know that,

$$\mathrm{cost}(x_j, V) \leq 2C_{i_*} = 2(C_L + C_L + C_{L-1} + C_{L-2} + \ldots + C_{i^*-1}) = 2C_L + \sum_{i=1}^{L} 2C_i\alpha_{ij}.$$

Moreover, $|M_i| = \alpha_{i1} + \ldots + \alpha_{in}$. So we have

$$\mathrm{cost}(X, V) = \sum_{i=1}^{n} \mathrm{cost}(x_j, V) \leq \sum_{j=1}^{n} \left(2C_L + \sum_{i=1}^{L} 2C_i\alpha_{ij}\right) = 2nC_L + 2\sum_{i=1}^{L} C_i|M_i| \leq \frac{C}{10} + 2\sum_{i=1}^{L} C_i|M_i|,$$

where the last inequality is due to the fact that $C_L = \frac{C_1}{2^{L-1}} \leq \frac{C_1}{20C_1 n/C} = \frac{C}{20n}$.

Then we have that,

$$\begin{aligned}
T =& \sum_{i=1}^{L} N_i C_i \\
\geq& \sum_{i=1}^{L} (|M_i| - CL^{-1}C_i^{-1}/8)C_i && \text{(Ineq. (A.1))} \\
=& \sum_{i=1}^{L} |M_i|C_i - C/8 \\
\geq& \text{cost}(X, x)/2 - C/20 - C/8 && \text{(Eq. (A.2))} \\
\geq& 5C - C/20 - C/8 && (\text{cost}(X, V) \geq 10C) \\
>& C.
\end{aligned}$$

Finally, we prove the second argument, i.e., if $\text{cost}(X, V) \leq C/10$, the algorithm rejects with probability at least $1-\delta$. Define sets $R_i = \{j \in [n] : \text{dist}(x_j, V) \geq C_i/2\}$, $Q_i = S_i \cap R_i$, and events $\mathcal{E}_i'' = \{\omega : \left|\frac{n}{S_i}|Q_i| - |R_i|\right| \geq CL^{-1}C_i^{-1}/8\}$ and $\mathcal{E}'' = \cap \mathcal{E}_i''$. By the same argument as in Lemma 2.2.2, we can prove that $\Pr[\mathcal{E} \cap \mathcal{E}''] \geq 1 - \delta$. Then we only need to show that if $\text{cost}(X, V) \leq C/10$, then $T < C$ conditioned on $\mathcal{E} \cap \mathcal{E}''$.

Conditioned on $\mathcal{E} \cap \mathcal{E}''$, if $C_i \leq C$ and $j \notin R_i$, we have

$$\text{dist}(x_j^{(i)}, V) \overset{\text{triangle ineq.}}{\leq} \text{dist}(x_j, V) + \|x_j^{(i)} - x_j\| \overset{j \notin R_i \text{ and } \mathcal{E}}{<} C_i/2 + r_i/2 \overset{r_i \leq C_i}{\leq} C_i.$$

It implies that $x_j \notin \{\mathcal{O}(x_l) \in S_i : |x^{(i)} - x_l^{(i)}| \geq C_i\}$. Hence we have $N_i \leq \frac{n}{|S_i|}|Q_i|$ in Line 9. Consequently,

$$N_i \leq \frac{n}{|S_i|}|Q_i| \overset{\mathcal{E}''}{\leq} |R_i| + CL^{-1}C_i^{-1}/8.$$

On the other hand, if $C_i > C$, recall that the algorithm checks every $j \in [n]$. Also note that if j contributes one to N_i then $j \in R_i$. We know that $N_i \leq |R_i|$ in this case. So we conclude that, for every i,

$$N_i \leq |R_i| + CL^{-1}C_i^{-1}/8. \tag{A.3}$$

Similar to Lemma 2.2.4, we claim the following conditioned on $\mathcal{E} \cap \mathcal{E}''$

$$\text{cost}(X, V) \geq \frac{1}{2} \sum_{i=1}^{L} C_i |R_i|. \tag{A.4}$$

To see this, for every $j \in [n]$, let β_{ij} denote the indicator function of $j \in R_i$. Fix a $j \in [n]$, let i^* be the smallest number such that $\beta_{ij} = 1$. If such i^* exists, by the definition of R_i and the fact that $C_{i+1} = C_i/2$,

$$\text{dist}(x_j, V) \geq C_{i^*}/2 > \frac{1}{2}(C_{i^*} + C_{i^*+1} + \ldots + C_L) = \frac{1}{2} \sum_{i=1}^{L} \beta_{ij} C_i.$$

Otherwise if such i^* does not exist, we have $\beta_{1j} = \ldots = \beta_{Lj} = 0$. Hence we have

$$\text{dist}(x_j, V) \geq 0 = \frac{1}{2} \sum_{i=1}^{L} \beta_{ij} C_i.$$

Thus, we always have $\text{dist}(x_j, V) \geq \frac{1}{2} \sum_{i=1}^{L} \beta_{ij} C_i$. Moreover, $|R_i| = \beta_{i1} + \beta_{i2} + \ldots + \beta_{iL}$. So we have,

$$\text{cost}(X, V) = \sum_{i=1}^{n} \text{dist}(x_j, V) \geq \sum_{j=1}^{n} \frac{1}{2} \left(\sum_{i=1}^{L} C_i \beta_{ij} \right) = \frac{1}{2} \sum_{i=1}^{L} C_i |R_i|.$$

Hence we have that,

$$\begin{aligned}
T =& \sum_{i=1}^{L} N_i C_i \\
\leq& \sum_{i=1}^{L} (|R_i| + CL^{-1} C_i^{-1}/8) C_i && \text{(Ineq. (A.3))} \\
=& \sum_{i=1}^{L} |R_i| C_i + C/8 && \text{(Ineq. (A.3))} \\
\leq& 2\text{cost}(X, x) + C/8 && \text{(Eq. (A.4))} \\
\leq& C/5 + C/8 && (\text{cost}(X, V) \leq C/10) \\
<& C.
\end{aligned}$$

■

Proof of Lemma 2.3.3. Let $V_i^{good} \subset V_i$ denote the set of $\frac{4|V_i|}{5}$ closest points to v_i. Then the probability that $\{x_j : j \in A_i\} \cap V_i^{good} = \emptyset$ is,

$$\begin{aligned}
(1 - \frac{|V_i^{good}|}{n})^{A_i} \leq & e^{-\frac{|V_i^{good}|A_i}{n}} \\
\leq & e^{-\frac{4|V_i|}{5} \cdot \Omega(\frac{n \log \delta^{-1}}{2^l}) \cdot \frac{1}{n}} \quad \text{(Defn. of } V_i^{good} \text{ and } A_i) \\
\leq & e^{-\frac{2^l \cdot 4}{5} \cdot \Omega(\frac{n \log \delta^{-1}}{2^l}) \cdot \frac{1}{n}} \qquad (|V_i| \geq 2^l) \\
\leq & \delta.
\end{aligned}$$

So with probability at least $1 - \delta$, we have a $j_1 \in A_i$ such that $x_{j_1} \in V_i^{good}$. That means

$$\begin{aligned}
\text{cost}(V_i, x_{j_1}) = \sum_{x_j \in V_i} \|x_j - x_{j_1}\| \leq & \sum_{x_j \in V_i} (\|v_i - x_j\| + \|v_i - x_{j_1}\|) \qquad \text{(triangle ineq.)} \\
= & \text{cost}(V_i, v_i) + |V_i| \|v_i - x_{j_1}\| \\
\leq & \text{cost}(V_i, v_i) + \frac{|V_i|}{|V_i| - |V_i^{good}|} \sum_{x_j \in V_i \setminus V_i^{good}} \|v_i - x_j\| \quad \text{(Defn. of } V_i^{good}) \\
\leq & \text{cost}(V_i, v_i) + \frac{|V_i|}{|V_i| - |V_i^{good}|} \text{cost}(V_i, v_i) \\
\leq & 6\text{cost}(V_i, v_i). \qquad (|V_i^{good}| = \frac{4|V_i|}{5})
\end{aligned}$$

■

proof of Lemma 2.3.4. Again, we condition on the event that every call of TestKmedian or NoisyKmedian succeeds, which happens with probability at least

$$1 - \frac{\delta}{10} - \sum_{i=1}^{+\infty} \frac{\delta}{10i(i+1)} \geq 1 - \delta,$$

by Lemma 2.3.2.

In Line 2, the first "IF" statement checks if OPT $\in \Omega(n)$. If it is the case, the

algorithm applies NoisyKmedian which takes

$$\tilde{O}_{\delta^{-1}}(d(n^3\mathrm{OPT}^{-2}+n)) = \tilde{O}_{\delta^{-1}}(dn)$$

many samples. The correctness is guaranteed by Theorem 2.1.3.

If $\mathrm{OPT} \in O(n)$, we obtain an $O(1)$-approximation clustering V when the algorithm terminates by Lemma 2.3.2 and a similar argument as in the proof of Theorem 2.2.6. The output C satisfies that $\mathrm{OPT}/10 \leq C \leq \mathrm{OPT}$ by Lemma 2.3.2 since TestK-NATKP$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, k, C, \frac{\delta}{10i(i+1)})$ rejects at the last round and TestK-NATKP$(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, k, 10C, \frac{\delta}{10(i-1)i})$ accepts at the penultimate round. Moreover, the sample complexity is determined by the last call of TestKmedian since C decreases exponentially, which takes

$$\tilde{O}_{\delta^{-1}}(dk^2(n^2\mathrm{OPT}^{-2}+n))$$

many samples by Lemma 2.3.2. ■

Appendix B

Coreset Construction

In the main text, we propose two algorithms NoisyKmedian and NoisyKmedian2 for computing the noisy k-median problem. However, the approximation factor seems quite large. In this section, we show how to improve the approximation factor arbitrarily close to 1. A simple way is to estimate the location of each point up to an error at most $\varepsilon\mathrm{OPT}/n$ and do the exhaustive search by the estimations. However, it requires $\tilde{O}_{\delta^{-1}}\left(d(n^3\mathrm{OPT}^{-2}\varepsilon^{-2}+n)\right)$ samples in total. We want to reduce this sample complexity. The main idea is to first construct a so-called ε-coreset (see Definition B.0.4) and do the exhaustive search on the coreset.

Definition B.0.4. (ε-coreset) Given a noisy k-median instance $X = \{\mathcal{O}(x_j) : j = 1, 2, \ldots, n\}$ in $\mathbb{R}^d$ and an $\varepsilon > 0$, an ε-coreset is defined to be a collection $\mathcal{S} \subset \mathbb{R}^d$ with a weight function $w : \mathcal{S} \to \mathbb{R}^+$ such that for any k centers V, the following property holds

$$\sum_{x\in\mathcal{S}} w(x)\mathrm{dist}(x, V) \in (1 \pm \varepsilon)\mathrm{cost}(X, V).$$

Note that an ε-coreset is a collection of deterministic points. Hence we can run any existing k-median algorithm on the coreset, e.g., BPRS or exhaustive search. The key problem is how to construct an ε-coreset. We propose Algorithm 8 to achieve this goal. We first run Algorithm NoisyKmedian to obtain an $O(1)$-approximate k-median clustering V and an $O(1)$-approximate k-median value D. Then we construct an ε-coreset by importance sampling. The construction approach is a combination

of Algorithm DisNATKP and Algorithm k-MEDIAN-CORESET in [41]. Note that in Line 3, we set the initial value r_0 to be $2D$ which is different from Algorithm DisNATKP. The reason is that D is a fixed value in Algorithm Coreset instead of an increasing value in Algorithm DisNATKP. Hence we let $r_0 = 2D$ such that $Tr_0 > \varepsilon D/100$ in Line 4. It ensures that Algorithm Coreset runs at least one iteration from Line 4 to Line 13 and estimates the locations of arms with high contribution. We estimate $\mathrm{dist}(x_j, V)$ for each $j \in [n]$ by the same technique as in Algorithm DisNATKP. From Line 4 to Line 19, we compute an upper bound $s(j)$ for the "importance" of each point x_j, i.e., $s(j) \geq \left\lceil \frac{n \cdot \mathrm{dist}(x_j, V)}{\mathrm{cost}(X,V)} \right\rceil + 1$; see Lemma B.0.5. This property guarantees the correctness of the importance sampling in Line 20, where we sample a collection U by the probability distribution determined by $s(j)$. Our ε-coreset contains the estimated locations of points in U and all points in V. The weight function is defined by the same way as in Algorithm k-MEDIAN-CORESET in [39].

Before proving the correctness of the ε-coreset, we need the following lemmas for preparation.

Lemma B.0.5. *With probability at least* $1 - \delta/4$, $s(j) \geq \left\lceil \frac{n \cdot \mathrm{dist}(x_j, V)}{\mathrm{cost}(X,V)} \right\rceil + 1$ *for each* $j \in [n]$. *Moreover,* $\sum_{j \in [n]} s(j) < 33n$.

Proof. By Theorem 2.1.3, we know that the output of $\mathsf{NoisyKmedian}(X, \delta/8)$ satisfies that $D \in [\frac{2}{5}, \frac{3}{2}] \cdot \mathrm{cost}(X, V)$ with probability at least $1 - \delta/8$. Let $G \subseteq [n]$ be the collection of j such that $\mathrm{flag}(j)$ =TRUE in Line 14. Observe that $|G| = n - T$ by Line 10. Define event $\mathcal{E} = \{\omega : \forall i \in \mathcal{Z}_+, j \in [n], \|x_j^{(i)} - x_j\| \leq r_i\}$. Note that $\Pr[\mathcal{E}] \geq 1 - \delta/8$. Conditioned on $\mathcal{E}$ and assume NoisyKmedian succeeds, whose probability is at least $1 - \delta/4$, we consider the following cases.

1) For any $j \in G$, assume that $s(j) = \left\lceil \frac{24nr_{i_j}}{D} \right\rceil + 1$ for some integer i_j. By Lemma 2.1.2, we have $r_{i_j} = r_{i_j - 1}/2 \geq \mathrm{dist}(x_j, V)/24$. Therefore, we have

$$s(j) = \left\lceil \frac{24nr_{i_j}}{D} \right\rceil + 1 \geq \left\lceil \frac{n \cdot \mathrm{dist}(x_j, V)}{\mathrm{cost}(X, V)} \right\rceil + 1.$$

2) If $j \notin G$, assume in Line 14, the variable $i = i^*$. By Lemma 2.1.2, we have

Algorithm 8 $\mathsf{Coreset}(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, \delta)$

Require: A sample access $\mathcal{O}(x_i)$ to $N(x_i, I_d)$ for each $i \in [n]$ and a confidence parameter $\delta \in (0, 1)$.

Ensure: An ε-coreset $\mathcal{S} \subset \mathbb{R}^d$ with a weight function $w : \mathcal{S} \to \mathbb{R}^+$.

Run $\mathsf{NoisyKmedian}(X, \delta/8)$. Let the output be a size-k set V and a value D.
For every $j \in [n]$, flag$(j) \leftarrow$FALSE. For every $v \in V$, $P_v \leftarrow \emptyset$.
$r_0 \leftarrow 2D$, $i \leftarrow 0$, $T \leftarrow n$, $\mathcal{S} \leftarrow \emptyset$, $t \leftarrow O\left(\varepsilon^{-2} k \min\{d \log d, \log k\} + k \log \frac{1}{\delta}\right)$, and $m \leftarrow O\left(\varepsilon^{-2} D^{-2} n^2 \log t\delta\right)$.
while $Tr_i \geq \varepsilon D/100$ **do**
  $r_{i+1} \leftarrow r_i/2$, $i \leftarrow i + 1$, and $m_i = O\big(r_i^{-2}(d + \log(\delta^{-1} n i))\big)$.
  Take m_i samples from every arm $\mathcal{O}(x_j)$ $(j \in [n])$ and compute the average $x_j^{(i)}$ as an estimate of x_j.
  for $j = 1, 2, \ldots, n$, flag(j) =FALSE **do**
    For every $v \in V$, compute $c_{jv} = \max_{y \in B(x_j^{(i)}, 3r_i)} \|y - v\|$ and $d_{jv} = \min_{y \in B(x_j^{(i)}, 3r_i)} \|y - v\|$.
    if $\big(\forall v \in V, d_{jv} > 0$ and $\exists v_1 \forall v_2 \neq v_1, c_{jv_1} \leq 2d_{jv_2}\big)$ **then**
      $P_{v_1} \leftarrow P_{v_1} \cup \{j\}$, proj$(j) \leftarrow \arg\min_{v \in V} \|x_j^{(i)} - v\|$, $s(j) \leftarrow \left\lceil \frac{24nr_i}{D} \right\rceil + 1$, flag$(j) \leftarrow$TRUE, $T \leftarrow T - 1$.
    end if
  end for
end while
for $j = 1, 2, \ldots, n$ **do**
  if flag(j) =FALSE **then**
    Compute $v \leftarrow \arg\min_{v' \in V} \|x_j^{(i)} - v'\|$.
    $P_v \leftarrow P_v \cup \{j\}$, proj$(j) \leftarrow v$, and $s(j) \leftarrow \left\lceil \frac{24nr_i}{D} \right\rceil + 1$.
  end if
end for
Pick a non-uniform random sample U of $[n]$ with probability $\frac{s(j)}{\sum_{j \in [n]} s(j)}$ for each $j \in [n]$.
for each $j \in U$ **do**
  Take m samples from arm $\mathcal{O}(x_j)$ and compute the average $\tilde{x}_j$ as an estimate of x_j.
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\tilde{x}_j\}$, and $w(\tilde{x}_j) \leftarrow \frac{\sum_{j \in [n]} s(j)}{t \cdot s(j)}$.
end for
for each $v \in V$ **do**
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$, and $w(v) \leftarrow (1 + 10\varepsilon)|P_v| - \sum_{j \in U \cap P_v} w(\tilde{x}_j)$.
end for
return $\mathcal{S}$ and w.

$r_{i^*} \geq \mathrm{dist}(x_j, V)/12$ Therefore, we have

$$s(j) = \left\lceil \frac{24nr_{i^*}}{D} \right\rceil + 1 \geq \left\lceil \frac{n \cdot \mathrm{dist}(x_j, V)}{\mathrm{cost}(X, V)} \right\rceil + 1.$$

Next, we prove $\sum_{j\in[n]} s(j) < 33n$. By Lemma 2.1.2, we have

$$r_{i_j} \leq \frac{1}{2}\mathrm{dist}(x_j, V) \tag{B.1}$$

for any $j \in G$, conditioned on $\mathcal{E}$. We have the following inequality

$$\begin{aligned}
\sum_{j\in[n]} s(j) &= \sum_{j\in G} s(j) + \sum_{j\in[n]\setminus G} s(j) \\
&= \sum_{j\in G} \left(\left\lceil \frac{24nr_{i_j}}{D} \right\rceil + 1 \right) + \sum_{j\in[n]\setminus G} \left(\left\lceil \frac{24nr_{i^*}}{D} \right\rceil + 1 \right) && \text{(Definitions of } s(j) \text{ and } i^*) \\
&\leq \sum_{j\in G} \left(\frac{24nr_{i_j}}{D} + 2 \right) + T \cdot \left(\frac{24nr_{i^*}}{D} + 2 \right) && \text{(Defn. of } T) \\
&\leq \sum_{j\in G} \frac{24nr_{i_j}}{0.4\mathrm{cost}(X, V)} + T \cdot \frac{24nr_{i^*}}{D} + 2n && (D \in [\frac{2}{5}, \frac{3}{2}] \cdot \mathrm{cost}(X, V)) \\
&\leq \sum_{j\in G} \frac{24n \cdot \frac{3}{2}\mathrm{dist}(x_j, V)}{0.4\mathrm{cost}(X, V)} + T \cdot \frac{24nr_{i^*}}{D} + 2n && \text{(Ineq. (B.1))} \\
&\leq \sum_{j\in G} \frac{30n\mathrm{dist}(x_j, V)}{\mathrm{cost}(X, V)} + \frac{24\varepsilon n}{100} + 2n && (Tr_{i^*} < \varepsilon D/100 \text{ by Line 4}) \\
&\leq 30n + 0.24n + 2n \\
&< 33n.
\end{aligned}$$

It completes the proof. ■

Lemma B.0.6. *With probability at least $1 - \delta/4$, we have $w(v) \geq 0$ for any $v \in V$.*

Proof. The lemma can be directly proved by the chosen of t and Corollary 15.3 in [41]. ■

Now we are ready to prove the main theorem of this section.

Theorem B.0.7. *With probability at least $1 - \delta$, $\mathsf{Coreset}(\{\mathcal{O}(x_i) : i = 1, 2, \ldots, n\}, \delta)$*

outputs an ε-coreset with sample complexity

$$\tilde{O}_{\delta^{-1}}\left(d(n^3\mathrm{OPT}^{-2}+\varepsilon^{-4}kn^2\mathrm{OPT}^{-2}+n)\right),$$

where $\mathrm{OPT}=\min_{y\in\mathbb{R}}\mathrm{cost}(X,y)$ *is the optimal* 1*-median value.*

Proof. Define event $\mathcal{E}_1=\left\{w:\mathrm{cost}(X,V)=O(1)\cdot\mathrm{OPT},D\in[\frac{2}{5},\frac{3}{2}]\cdot\mathrm{cost}(X,V)\right\}$. By Theorem 2.1.3, we have $\Pr[\mathcal{E}_1]\geq 1-\delta/4$. Define $\mathcal{M}$ to be the collection of all k center sets in $\mathbb{R}^d$. For every $j\in[n]$, let $l_j:\mathcal{M}\to\mathbb{R}^+$ be defined as follows:

$$l_j(S)=\frac{\mathrm{dist}(x_j,S)-\mathrm{dist}(\mathrm{proj}(j),S)}{s(j)}+\frac{100\cdot\mathrm{cost}(X,V)}{\sum_{j\in[n]}s(j)}.$$

Observe that the above function is similar to Definition 14.4 in [39]. Define $\mathcal{E}_2=\{\omega:\forall v\in\mathcal{S},w(v)\geq 0\}$. By Lemma B.0.6, we have $\Pr[\mathcal{E}_2]\geq 1-\delta/4$. Conditioned on $\mathcal{E}_2$, we claim that with probability at least $1-\delta/4$, for any $S\in\mathcal{M}$, the following property holds:

$$\left|\mathrm{cost}(X,S)-\left(\sum_{j\in[n]}\mathrm{dist}(\mathrm{proj}(j),S)+\sum_{j\in U}w(\tilde{x}_j)\mathrm{dist}(x_j,S)-\sum_{j\in U}w(\tilde{x}_j)\mathrm{dist}(\mathrm{proj}(j),S)\right)\right| \tag{B.2}$$
$$\leq\varepsilon\mathrm{cost}(X,B).$$

Note that we set $t\leftarrow O\left(\varepsilon^{-2}k\min\{d\log d,\log k\}+k\log\frac{1}{\delta}\right)$, $w(\tilde{x}_j)\leftarrow\frac{\sum_{j\in[n]}s(j)}{t\cdot s(j)}$, and $s(j)$ satisfying Lemma B.0.5. Combining the proof of Theorems 14.5 and 16.4 in [41], we only need to prove that $l_j(S)\geq 0$ for any $j\in[n]$ and $S\in\mathcal{M}$, with probability at least $1-\delta/8$.

Define events $\mathcal{E}_i=\{\omega:\forall j\in[n],\|x_j^{(i)}-x_j\|\leq r_i\}$ and $\mathcal{E}=\cap_{i\geq 1}\mathcal{E}_i$. By the proof of Lemma 2.1.1, we know that $\Pr[\mathcal{E}]\geq 1-\delta/8$. By the triangle inequality, for any $j\in[n]$ and $S\in\mathcal{M}$, we have

$$|\mathrm{dist}(x_j,S)-\mathrm{dist}(\mathrm{proj}(j),S)|\leq\mathrm{dist}(x_j,\mathrm{proj}(j)).$$

Thus it suffices to prove that

$$\mathrm{dist}(x_j, \mathrm{proj}(j)) \leq \frac{100s(j) \cdot \mathrm{cost}(X, V)}{\sum_{j \in [n]} s(j)} \tag{B.3}$$

conditioned on $\mathcal{E}$. Let $G \subseteq [n]$ be the collection of j such that flag(j) =TRUE in Line 14. Observe that $|G| = n - T$ by Line 10. We discuss the following two cases.

1) If $j \in G$, assume that flag(j) is set to be TRUE at the i_j-th round. We have the following

$$\begin{aligned}
\mathrm{dist}(x_j, \mathrm{proj}(j)) &\leq \mathrm{dist}(x_j, x_j^{(i_j)}) + \mathrm{dist}(x_j^{(i_j)}, \mathrm{proj}(j)) && \text{(triangle ineq.)} \\
&\leq r_{i_j} + \mathrm{dist}(x_j^{(i_j)}, \mathrm{proj}(j)) && (\mathcal{E}) \\
&= r_{i_j} + \mathrm{dist}(x_j^{(i_j)}, V) && \text{(Defn. of proj}(j)) \\
&\leq \frac{1}{2}\mathrm{dist}(x_j, V) + \frac{3}{2}\mathrm{dist}(x_j, V) && \text{(Lemma 2.1.2)} \\
&= 2\mathrm{dist}(x_j, V) \\
&= \frac{66 \cdot \frac{n \cdot \mathrm{dist}(x_j, V)}{\mathrm{cost}(X, V)} \cdot \mathrm{cost}(X, V)}{33n} \\
&\leq \frac{100s(j) \cdot \mathrm{cost}(X, V)}{\sum_{j \in [n]} s(j)}. && \text{(Lemma B.0.5)}
\end{aligned}$$

2) If $j \notin G$, assume in Line 14, the variable $i = i^*$. We have the following

$$\begin{aligned}
\mathrm{dist}(x_j, \mathrm{proj}(j)) &\le \mathrm{dist}(x_j, x_j^{(i^*)}) + \mathrm{dist}(x_j^{(i^*)}, \mathrm{proj}(j)) && \text{(triangle ineq.)} \\
&\le r_{i^*} + \mathrm{dist}(x_j^{(i^*)}, \mathrm{proj}(j)) && (\mathcal{E}) \\
&= r_{i^*} + \mathrm{dist}(x_j^{(i^*)}, V) && \text{(Defn. of proj}(j)) \\
&\le r_{i^*} + \mathrm{dist}(x_j, x_j^{(i^*)}) + \mathrm{dist}(x_j, V) && \text{(triangle ineq.)} \\
&\le 2r_{i^*} + \mathrm{dist}(x_j, V) && (\mathcal{E}) \\
&\le 2r_{i^*} + 12r_{i^*} && \text{(Lemma 2.1.2)} \\
&\le \frac{66 \cdot \frac{7nr_{i^*}}{\mathrm{cost}(X,V)} \cdot \mathrm{cost}(X,V)}{33n} \\
&\le \frac{66 \cdot \frac{20nr_{i^*}}{D} \cdot \mathrm{cost}(X,V)}{33n} && (\mathcal{E}_1) \\
&\le \frac{100s(j) \cdot \mathrm{cost}(X,V)}{\sum_{j \in [n]} s(j)}. && \text{(Defn. of } s(j) \text{ and Lemma B.0.5)}
\end{aligned}$$

Next, we define event $\mathcal{E}_3 = \{\omega : \forall j \in [n], \|\tilde{x}_j - x_j\| \le \frac{\varepsilon D}{50n}\}$. Similar to the proof of Lemma 2.1.1, we have $\Pr[\mathcal{E}_3] \ge 1 - \delta/4$. Define event $\mathcal{E} = \{w : \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ and Claim (B.2) succeeds$\}$. By the union bound, we have $\Pr[\mathcal{E}] \ge 1 - \delta$. It suffices to prove that $\mathcal{S}$ is an ε-coreset conditioned on $\mathcal{E}$.

For any $S \in \mathcal{M}$, we have

$$\begin{aligned}
&\left|\sum_{j\in U} w(\tilde{x}_j)\mathrm{dist}(\tilde{x}_j, S) - \sum_{j\in U} w(\tilde{x}_j)\mathrm{dist}(x_j, S)\right| = \left|\sum_{j\in U} w(\tilde{x}_j)\left(\mathrm{dist}(\tilde{x}_j, S) - \mathrm{dist}(x_j, S)\right)\right| \\
\leq &\left|\sum_{j\in U} w(\tilde{x}_j)\|\tilde{x}_j - x_j\|\right| && \text{(triangle ineq.)} \\
\leq &\frac{\varepsilon D}{50n}\left|\sum_{j\in U} w(\tilde{x}_j)\right| && (\mathcal{E}_3) \\
= &\frac{\varepsilon D}{50n}\left|\sum_{j\in U} \frac{\sum_{j\in[n]} s(j)}{t\cdot s(j)}\right| && \text{(Defn. of } w(\tilde{x}_j)\text{)} \\
\leq &\frac{\varepsilon D}{50n}\left|\sum_{j\in U} \frac{\sum_{j\in[n]} s(j)}{t}\right| && (s(j)\geq 1) \\
\leq &\frac{\varepsilon D}{50n}\cdot 33n && \text{(Lemma B.0.5)} \\
= &O(\varepsilon)\cdot \mathrm{OPT} && (\mathcal{E}_1) \\
\leq &O(\varepsilon)\cdot \mathrm{cost}(X, S). && \text{(Defn. of OPT)}
\end{aligned} \tag{B.4}$$

We also have

$$\begin{aligned}
&\left|\mathrm{cost}(X,S) - \sum_{j\in U} w(\tilde{x}_j)\mathrm{dist}(x_j,S) - \sum_{v\in V} w(v)\mathrm{dist}(v,S)\right| \\
= &\left|\mathrm{cost}(X,S) - \sum_{j\in U} w(\tilde{x}_j)\mathrm{dist}(x_j,S) - \sum_{v\in V}\left((1+10\varepsilon)|P_v| - \sum_{j\in U\cap P_v} w(\tilde{x}_j)\right)\mathrm{dist}(v,S)\right| \\
\leq &\left|\mathrm{cost}(X,S) - \sum_{j\in U} w(\tilde{x}_j)\mathrm{dist}(x_j,S) - \sum_{v\in V}\left(|P_v| - \sum_{j\in U\cap P_v} w(\tilde{x}_j)\right)\mathrm{dist}(v,S)\right| \\
&+ 10\varepsilon\left|\sum_{v\in V}|P_v|\cdot\mathrm{dist}(v,S)\right| \qquad \text{(triangle ineq.)} \\
= &\left|\mathrm{cost}(X,S) - \left(\sum_{j\in[n]}\mathrm{dist}(\mathrm{proj}(j),S) + \sum_{j\in U} w(\tilde{x}_j)\mathrm{dist}(x_j,S) - \sum_{j\in U} w(\tilde{x}_j)\mathrm{dist}(\mathrm{proj}(j),S)\right)\right| \\
&+ 10\varepsilon\sum_{v\in V}|P_v|\cdot\mathrm{dist}(v,S) \qquad (\mathrm{proj}(j) = v \text{ if } j\in P_v) \\
\leq &\varepsilon\mathrm{cost}(X,S) + 10\varepsilon\sum_{v\in V}|P_v|\cdot\mathrm{dist}(v,S) \qquad \text{(Claim (B.2))}.
\end{aligned} \tag{B.5}$$

Combining Inequalities (B.4) and (B.5), we have

$$\begin{aligned}
&\left|\mathrm{cost}(X,S)-\sum_{v\in\mathcal{S}}w(v)\mathrm{dist}(v,S)\right| \le \left|\sum_{j\in U}w(\tilde{x}_j)\mathrm{dist}(\tilde{x}_j,S)-\sum_{j\in U}w(\tilde{x}_j)\mathrm{dist}(x_j,S)\right| \\
&+\left|\mathrm{cost}(X,S)-\sum_{j\in U}w(\tilde{x}_j)\mathrm{dist}(x_j,S)-\sum_{v\in V}w(v)\mathrm{dist}(v,S)\right| && \text{(triangle i} \\
\le& O(\varepsilon)\cdot\mathrm{cost}(X,S)+10\varepsilon\sum_{v\in V}|P_v|\cdot\mathrm{dist}(v,S). && \text{(Ineq. (B.4) and (}
\end{aligned}$$

It remains to show that $10\varepsilon\sum_{v\in V}|P_v|\cdot\mathrm{dist}(v,S)=O(\varepsilon)\cdot\mathrm{cost}(X,S)$ which implies that $\mathcal{S}$ is an $O(\varepsilon)$-coreset conditioned on $\mathcal{E}$. Then we have the following

$$\begin{aligned}
&10\varepsilon\sum_{v\in V}|P_v|\cdot\mathrm{dist}(v,S) \\
=&10\varepsilon\sum_{j\in[n]}\mathrm{dist}(\mathrm{proj}(j),S) && \text{(Defn. of proj}(j)) \\
\le&10\varepsilon\sum_{j\in[n]}(\mathrm{dist}(x_j,S)+\mathrm{dist}(x_j,\mathrm{proj}(j))) && \text{(triangle ineq.)} \\
\le&10\varepsilon\cdot\mathrm{cost}(X,S)+10\varepsilon\sum_{j\in[n]}\frac{100s(j)\cdot\mathrm{cost}(X,V)}{\sum_{j\in[n]}s(j)} && \text{(Ineq. (B.3))} \\
=&10\varepsilon\cdot\mathrm{cost}(X,S)+1000\varepsilon\cdot\mathrm{cost}(X,V) \\
=&10\varepsilon\cdot\mathrm{cost}(X,S)+O(\varepsilon)\cdot\mathrm{OPT} && (\mathcal{E}_1) \\
\le&O(\varepsilon)\cdot\mathrm{cost}(X,S), && \text{(Defn. of OPT)}
\end{aligned}$$

which completes the proof of the correctness.

Finally, we prove the sample complexity. By Theorem 2.1.3, the number of samples for NoisyKmedian is $\tilde{O}_{\delta^{-1}}(d(n^3\mathrm{OPT}^{-2}+n))$. Since the process from Line 4 to Line 13 is almost identical to Algorithm DisNATKP and $D=O(\mathrm{OPT})$, we can prove that the number of samples is $\tilde{O}_{\delta^{-1}}(d(n^3\mathrm{OPT}^{-2}+n))$ by the same argument as in Lemma 2.1.1. From Line 21 to Line 24, the total number of samples is $mt=\tilde{O}_{\delta^{-1}}\left(d\varepsilon^{-4}kn^2\mathrm{OPT}^{-2}\right)$. It completes the proof. ■

Bibliography

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. In *STOC*, pages 271–286. ACM, 2006.

[2] Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Approximating extent measures of points. *Journal of the ACM (JACM)*, 51(4):606–635, 2004.

[3] Pankaj K Agarwal and Cecilia Magdalena Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.

[4] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing low-degree polynomials over gf (2). In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pages 188–199. Springer, 2003.

[5] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.

[6] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, pages 1027–1035, 2007.

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

[8] P. Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[9] Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable and distributed clustering via lightweight coresets. *CoRR*, abs/1702.08248, 2017.

[10] Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The traveling salesman problem: Low-dimensionality implies a polynomial time approximation scheme. *SIAM J. Comput.*, 45(4):1563–1581, 2016.

[11] Mihir Bellare, Don Coppersmith, JOHAN Hastad, Marcos Kiwi, and Madhu Sudan. Linearity testing in characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, 1996.

[12] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of computer and system sciences*, 47(3):549–595, 1993.

[13] Vladimir Braverman, Dan Feldman, and Harry Lang. New frameworks for offline and streaming coreset constructions. *CoRR*, abs/1612.00889, 2016.

[14] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 2012.

[15] Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Transactions on Algorithms*, 13:1–31, 03 2017.

[16] Wei Cao, Jian Li, Yufei Tao, and Zhize Li. On top-k selection in multi-armed bandits and hidden bipartite graphs. In *NIPS*, pages 1036–1044, 2015.

[17] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[18] T.-H. Hubert Chan and Khaled M. Elbassioni. A QPTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics. *Discrete & Computational Geometry*, 46(4):704–723, 2011.

[19] T.-H. Hubert Chan and Anupam Gupta. Small hop-diameter sparse spanners for doubling metrics. *Discrete & Computational Geometry*, 41(1):28–44, 2009.

[20] T.-H. Hubert Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. *ACM Trans. Algorithms*, 12(4):55:1–55:22, 2016.

[21] T.-H. Hubert Chan, Anupam Gupta, and Kunal Talwar. Ultra-low-dimensional embeddings for doubling metrics. *J. ACM*, 57(4):21:1–21:26, 2010.

[22] T.-H. Hubert Chan, Shuguang Hu, and Shaofeng H.-C. Jiang. A PTAS for the steiner forest problem in doubling metrics. In *FOCS*, pages 810–819. IEEE Computer Society, 2016.

[23] T.-H. Hubert Chan and Shaofeng H.-C. Jiang. Reducing curse of dimensionality: Improved PTAS for TSP (with neighborhoods) in doubling metrics. *ACM Trans. Algorithms*, 14(1):9:1–9:18, 2018.

[24] T.-H. Hubert Chan, Mingfei Li, and Li Ning. Sparse fault-tolerant spanners for doubling metrics with bounded hop-diameter or degree. *Algorithmica*, 71(1):53–65, 2015.

[25] T.-H. Hubert Chan, Mingfei Li, Li Ning, and Shay Solomon. New doubling spanners: Better and simpler. *SIAM J. Comput.*, 44(1):37–53, 2015.

[26] Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002.

[27] 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.

[28] Ke Chen. On k-median clustering in high dimensions. In *SODA*, pages 1177–1185. Society for Industrial and Applied Mathematics, 2006.

[29] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *NIPS*, pages 379–387, 2014.

[30] Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond talagrand functions: new lower bounds for testing monotonicity and unateness. *arXiv preprint arXiv:1702.06997*, 2017.

[31] K. L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete Comput. Geom.*, 22(1):63–93, 1999.

[32] Adam Coates and Andrew Y. Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade - Second Edition*, pages 561–580. 2012.

[33] Richard Cole and Lee-Ad Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *38thSTOC*, pages 574–583, 2006.

[34] Graham Cormode and Andrew McGregor. Approximation algorithms for clustering uncertain data. In *Proceedings of the twenty-seventh ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 191–200. ACM, 2008.

[35] Roee David, Irit Dinur, Elazar Goldenberg, Guy Kindler, and Igor Shinkar. Direct sum testing. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 327–336. ACM, 2015.

[36] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *FOCS*, pages 73–84.

[37] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*, pages 97–108. Springer, 1999.

[38] Funda Ergün, Sampath Kannan, S Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *Journal of Computer and System Sciences*, 60(3):717–751, 2000.

[39] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *STOC*, pages 569–578, 2011.

[40] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578. ACM, 2011.

[41] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *STOC*, pages 569–578, 2011.

[42] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In *SODA*, pages 1434–1453, 2013.

[43] Dan Feldman and Leonard J Schulman. Data reduction for weighted and outlier-resistant clustering. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 1343–1354. Society for Industrial and Applied Mathematics, 2012.

[44] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 474–483. ACM, 2002.

[45] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. In *FOCS*, pages 365–374. IEEE Computer Society, 2016.

[46] Jie Gao, Leonidas J. Guibas, and An Thanh Nguyen. Deformable spanners and applications. In *Symposium on Computational Geometry*, pages 190–199. ACM, 2004.

[47] Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

[48] Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 33. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[49] Oded Goldreich and Or Sheffet. On the randomness complexity of property testing. *computational complexity*, 19(1):99–133, 2010.

[50] Lee-Ad Gottlieb and Liam Roditty. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *SODA*, pages 591–600. SIAM, 2008.

[51] Lee-Ad Gottlieb and Liam Roditty. An optimal dynamic spanner for doubling metric spaces. In *ESA*, volume 5193 of *Lecture Notes in Computer Science*, pages 478–489. Springer, 2008.

[52] Sudipto Guha and Kamesh Munagala. Exceeding expectations and clustering uncertain data. In *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 269–278. ACM, 2009.

[53] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS*, pages 534–543. IEEE Computer Society, 2003.

[54] Sariel Har-Peled. Clustering motion. *Discrete & Computational Geometry*, 31(4):545–565, 2004.

[55] Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.

[56] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300. ACM, 2004.

[57] Sariel Har-Peled and Manor Mendel. Fast construction of nets in low dimensional metrics, and their applications. In *Symposium on Computational Geometry*, pages 150–158. ACM, 2005.

[58] Lingxiao Huang, Shaofeng H.-C Jiang, Jian Li, and Xuan Wu. ε-coresets for clustering(with outliers) in doubing metrics. *CoRR*, abs/1804.02530, 2018.

[59] Lingxiao Huang and Jian Li. Stochastic k-center and j-flat-center problems. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 110–129. Society for Industrial and Applied Mathematics, 2017.

[60] Piotr Indyk and Assaf Naor. Nearest neighbor preserving embeddings. *ACM Transactions on Algorithms*, To appear.

[61] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 731–740. ACM, 2002.

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

[63] Charanjit S Jutla, Anindya C Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 423–432. IEEE, 2004.

[64] Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM Journal on Computing*, 36(3):779–802, 2006.

[65] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

[66] Michael J Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. MIT press, 1994.

[67] Michael Langberg and Leonard J. Schulman. Universal ϵ-approximators for integrals. In *SODA*, pages 598–607, 2010.

[68] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM Journal on Computing*, 45(2):530–547, 2016.

[69] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *J. Comput. Syst. Sci.*, 62(3):516–527, 2001.

[70] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28(2):129–136, 1982.

[71] Mario Lucic, Matthew Faulkner, Andreas Krause, and Dan Feldman. Training mixture models at scale via coresets. *arXiv preprint arXiv:1703.08110*, 2017.

[72] Arya Mazumdar and Barna Saha. Clustering with noisy queries. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5788–5799. Curran Associates, Inc., 2017.

[73] Jeff M. Phillips. Coresets and sketches. *CoRR*, abs/1601.00617, 2016.

[74] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 1997.

[75] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[76] Amir Shpilka and Avi Wigderson. Derandomizing homomorphism testing in general groups. *SIAM Journal on Computing*, 36(4):1215–1230, 2006.

[77] Shay Solomon. From hierarchical partitions to hierarchical covers: optimal fault-tolerant spanners for doubling metrics. In *STOC*, pages 363–372. ACM, 2014.

[78] Talagrand. *Upper and Lower Bound of Stochastic Process: Modern Methods and Classical Problems*. Springer, 2014.

[79] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *STOC*, pages 281–290. ACM, 2004.

[80] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, et al. Cluster analysis: basic concepts and algorithms. *Introduction to data mining*, 8:487–568, 2006.

[81] Ramon van Handel. Probability in high dimension. Technical report, DTIC Document, 2014.

[82] Haitao Wang and Jingru Zhang. One-dimensional k-center on uncertain data. *Theoretical Computer Science*, 602:114–124, 2015.

[83] Yuan Zhou, Xi Chen, and Jian Li. Optimal pac multiple arm identification with applications to crowdsourcing. In *ICML*, pages 217–225, 2014.