NORTHWESTERN UNIVERSITY

Approximation Algorithms for Explainable Clustering

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Liren Shan

EVANSTON, ILLINOIS

September 2023

ABSTRACT

Clustering is a fundamental task in unsupervised learning, which aims to partition the data set into several clusters. It is widely used for data mining, image segmentation, and natural language processing. One of the most popular clustering methods is centroid-based clustering, including k-medians and k-means clustering. k-medians and k-means clustering choose k centers and assign each data point to its closest center. Thus, this clustering forms a Voronoi partition of the space based on k centers. Each cluster corresponds to a Voronoi cell, which usually has a complicated boundary. Hence, it is not necessarily easy for humans to understand these clusters. In real-world applications, many important decisions are made for different clusters created by clustering algorithms. To make these decisions more interpretable, we want to find more explainable clustering.

In this thesis, we study approximation algorithms for explainable k-medians and k-means clustering. The problem of explainable k-medians and k-means was recently introduced by Dasgupta, Frost, Moshkovitz, and Rashtchian (ICML 2020). For this problem, our goal is to find a threshold decision tree that partitions data into k clusters and minimizes the k-medians or k-means objective. The obtained clustering is easy to interpret because every decision node of a threshold tree splits the node into two groups with a threshold cut on a single feature. The price of explainability is defined as the ratio of its cost and the optimal unconstrained cost. We provide an efficient algorithm that achieves the optimal and near-optimal upper bounds on the price of explainability for k-medians in ℓ_1 and k-means, respectively. We also provide a competitive algorithm and lower bound for explainable k-medians in ℓ_2. Finally, we provide a bi-criteria competitive algorithm that creates a k clustering by using a threshold tree with slightly more than k leaves. We show an exponential improvement in the price of explainability for k-means by adding a constant fraction of extra leaves. This captures the tradeoff between accuracy and explainability.

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Konstantin Makarychev. Kostya guides me through these five years of my Ph.D. study not only as a great advisor but also as a close friend. We spent so much wonderful time in his office discussing interesting problems. During the tough time of COVID, when I found the exciting work on explainable clustering by Dasgupta, Frost, Moshkovitz, and Rashtchian (2020), he encouraged me to present it in his online course. Through our Zoom meetings, he gave me many valuable intuitions and amazing ideas about this problem. I can not imagine how hard it would be to get the results of this thesis without his guidance.

I would like to thank my committee members, Aravindan Vijayaraghavan, Jason Hartline, and Yury Makarychev. They provided great suggestions for my research and this thesis. I am also fortunate to collaborate with them on research projects. I really enjoy interesting courses by Jason: mechanism design and CS+Law, which broaden my research interests and provide me with useful research techniques. From the projects with Jason and other collaborators, I learn how to formulate an interesting model to capture real-world problems. I also benefit a lot from the buffered graph partitioning project with Kostya, Yury, and Aravindan. Yury and Aravindan provide many significant insights into this problem, which eventually lead to our interesting results.

I want to thank my collaborators during my Ph.D. study: Charlie Carlson, Jafar Jafarov, Karl Johansson, Yingkai Li, Dan Linna, Edmond Lou, Philip Paré, Aravind Reddy, Alex Tang, Yifan Wu, Yuhao Yi. With their help and support, we complete all those interesting research on a wide range of topics including clustering, graph partitioning, mechanism design, control, online learning, and law. It is a great pleasure to work with them to explore these problems and learn a lot from their expertise.

I would also like to thank all members of the Northwestern CS Theory group, which is like

a warm family. I first thank all faculties and postdocs for their amazing courses and support: Anindya De, Ben Golub, Samir Khuller, Annie Liang, Xiao Wang, Huck Bennett, Hedyeh Beyhaghi, Xue Chen, Vaggos Chatziafratis, Sami Davies, Jinshuo Dong, Quanquan C. Liu, Shravas Rao, Don Stull. I also want to thank all my colleagues and friends: Yiding Feng, Abhratanu Dutta, Aleck Johnsen, Paula Kayongo, Leif Rasmussen, Aidao Chen, Charles Cui, Nirmit Joshi, Sanchit Kalhan, Yiduo Ke, Sheng Long, Michail Mamakos, Anant Shah, Vaidehi Srinivas, Pattara Sukprasert, Matthew VonAllmen, Sheng Yang, Chenhao Zhang, Shirley Zhang, Liam O'Carroll, and Nathan White.

I am also grateful for taking fantastic courses from all faculties and postdocs: Aditya Bhaskara, Randall Berry, Eddie Dekel, Jorge Nocedal, Zhaoran Wang, Steven Zelditch, Antonio Auffinger, Erin Leddon. I learned techniques, intuitions, and high-level ideas on various topics from classic materials to modern research from their courses. I had a great internship at Yahoo Research. I would like to thank the whole Yahoo Research team and my mentors there: Balazs Szorenyi, Mahdi Khalili, Maxim Sviridenko. I thank all faculties and friends I worked with during my undergrad for leading me into the research path: Zhongzhi Zhang, Richard Peng, Huan Li, Guanrong Chen, Yujia Jin. I want to thank all staff at Northwestern for their support and services: Xiaolin Wang, Julia Blend, Wynante Charles, Jensen Smith, Katie Winters, Keziah Tetteh, James Devine, Tatiana Meza.

Finally, I want to thank my parents and other family members. I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Clustering is a fundamental task in data analysis. The goal of clustering is to partition a data set into several clusters such that similar data points are in the same cluster. Clustering is used in many fields, including bioinformatics, medicine, engineering, and business. They use clustering algorithms to discover the hidden pattern inside the data. Many important decisions are then made based on the hidden pattern learned by the clustering algorithm. Different decisions can be picked for clusters partitioned by the clustering algorithm. To make these decisions more interpretable, we want to find an *explainable* clustering – clustering which can be easily understood by a human being.

One commonly used clustering method is centroid-based clustering, which includes popular k-means and k-medians clustering. Given a set of data points X in $\mathbb{R}^d$, for k-means or k-medians clustering, we need to find k centers and then assign each data point to its closest center. Specifically, k-means or k-medians clustering forms a d-dimensional Voronoi diagram for centers $c^1, c^2, \ldots, c^k$, in which, the i-th cluster P_i contains those points in X that are closer to c^i than to any other center c^j. The k-medians and k-means problems are to find a set C of k centers $c^1, c^2, \cdots, c^k$ to minimize the corresponding costs: k-medians in ℓ_1 cost (1.1), k-medians in ℓ_2 cost (1.2), and k-means cost (1.3).

$$\mathrm{cost}_{\ell_1}(X, C) = \sum_{i=1}^{d} \sum_{x \in P_i} \|x - c^i\|_1, \tag{1.1}$$

$$\mathrm{cost}_{\ell_2}(X, C) = \sum_{i=1}^{d} \sum_{x \in P_i} \|x - c^i\|_2. \tag{1.2}$$

$$\text{cost}_{\ell_2^2}(X, C) = \sum_{i=1}^{d} \sum_{x \in P_i} \|x - c^i\|_2^2. \tag{1.3}$$

where P_i is the i-th cluster.

Although every cluster in a k-means and k-medians clustering has a simple mathematical description, this description is not necessarily easy to interpret for a human. In order to determine to which cluster a particular point belongs, we need to compute distances from point x to all centers c^j. Each distance depends on all coordinates of the points. Hence, for a human, it is not even easy to figure out to which cluster in k-means or k-medians clustering a particular point belongs to; let alone interpret the entire clustering.

In everyday life, we are surrounded by different types of classifications. Consider the following examples from Wikipedia: *(1) Performance cars are capable of going from 0 to 60 mph in under 5 seconds*; *(2) Modern sources currently define skyscrapers as being at least 100 meters or 150 meters in height; (3) Very-low-calorie diets are diets of 800 kcal or less energy intake per day, whereas low-calorie diets are between 1000-1200 kcal per day*. Note that all these definitions depend on a *single feature* which makes them easy to understand.

In a recent ICML paper, Dasgupta, Frost, Moshkovitz, and Rashtchian (2020) proposed to use a threshold decision tree to create a clustering with concise explanations of clusters. A threshold tree is a binary classification tree with k leaves. Every internal node u of the tree splits the data into two sets by comparing a single feature i_u of each data point with a threshold θ_u. The first set is the set of points with $x_{i_u} \leq \theta_u$; the second set is the set of points with $x_{i_u} > \theta_u$. These two sets are then recursively partitioned by the left and right children of u. Thus, each point x in the data set is eventually assigned to one of k leaves of the threshold tree $\mathcal{T}$. This gives us a partitioning of the data set X into clusters $\mathcal{P} = (P_1, \ldots, P_k)$. We note that threshold decision trees are special cases of binary space partitioning (BSP) trees and similar to k-d trees (Bentley, 1975).

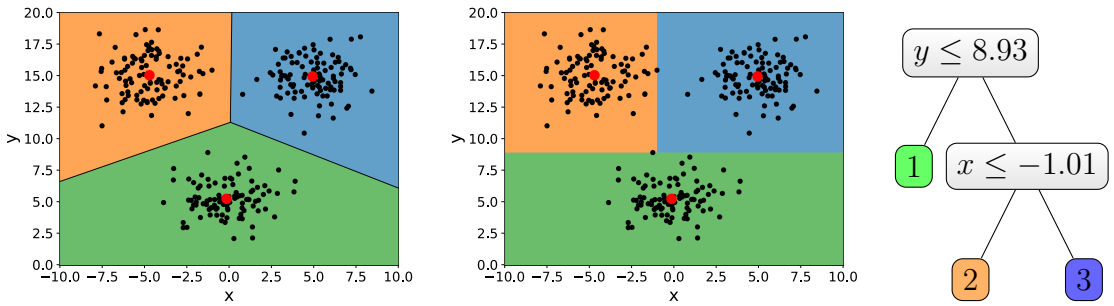


Figure 1.1: Explainable and non-explainable k-means. The left diagram shows the optimal Voronoi partition of the plane. The middle diagram shows an explainable partition. The right diagram shows the corresponding decision tree for explainable clustering.

Dasgupta et al. (2020) suggested that we measure the quality of a threshold tree using the standard k-means and k-medians objectives. Specifically, the k-medians in ℓ_1 cost of the threshold tree $\mathcal{T}$ equals (1.4), the k-medians in ℓ_2 cost equals (1.5) and k-means cost equals (1.6):

$$\text{cost}_{\ell_1}(X, \mathcal{T}) = \sum_{i=1}^{k} \sum_{x \in P_i} \|x - c^i\|_1, \tag{1.4}$$

$$\text{cost}_{\ell_2}(X, \mathcal{T}) = \sum_{i=1}^{k} \sum_{x \in P_i} \|x - c^i\|_2, \tag{1.5}$$

$$\text{cost}_{\ell_2^2}(X, \mathcal{T}) = \sum_{i=1}^{k} \sum_{x \in P_i} \|x - c^i\|_2^2, \tag{1.6}$$

where c^i is the ℓ_1-median of cluster P_i in (1.4), the ℓ_2-median of cluster P_i in (1.5), and the mean of cluster P_i in (1.6).

This definition raises obvious questions: Can we actually find a good explainable clustering? Moreover, how good can it be comparing to an unconstrained k-medians and k-means clustering? Let $\text{OPT}_{\ell_1}(X)$, $\text{OPT}_{\ell_2}(X)$, and $\text{OPT}_{\ell_2^2}(X)$ be the optimal solutions to unconstrained k-medians in ℓ_1, k-medians in ℓ_2, and k-means, respectively. Dasgupta et al. (2020) defined

the *price of explainability* for an explainable clustering given by decision tree $\mathcal{T}$ as the ratio $\text{cost}_{\ell_1}(X, \mathcal{T})/\,\text{OPT}_{\ell_1}(X)$ for k-medians in ℓ_1 and $\text{cost}_{\ell_2^2}(X, T)/\,\text{OPT}_{\ell_2^2}(X)$ for k-means. The price of explainability shows by how much the optimal unconstrained solution is better than the explainable solution for the same data set.

In their paper, Dasgupta et al. (2020) gave upper and lower bounds on the price of explainability. They proved that the price of explainability is upper bounded by $O(k)$ and $O(k^2)$ for k-medians in ℓ_1 and k-means, respectively. The cost of explainability for k-medians in ℓ_1 and k-means (somewhat surprisingly) does not depend on the number of points in the data set X and only depends on the number of centers k. Specifically, they provided a greedy algorithm that given k reference centers $c^1, c^2, \cdots, c^k$ of any unconstrained k-medians in ℓ_1 clustering as input, outputs a threshold decision tree of cost at most $O(k)$ times the cost of original unconstrained k-medians clustering with centers $c^1, c^2, \cdots, c^k$. We call such an algorithm $O(k)$ competitive. To get an explainable k-medians in ℓ_1 clustering, we first obtain reference centers $c^1, c^2, \cdots, c^k$ using an off-the-shelf approximation algorithm for k-medians in ℓ_1 and then run an α-competitive algorithm for explainable k-medians with centers $c^1, c^2, \cdots, c^k$ given as input. This algorithm produces the desired threshold decision tree. Dasgupta et al. (2020) also showed that this greedy algorithm is $O(k^2)$ competitive for explainable k-means and showed $\Omega(\log k)$ lower bounds on the price of explainability for both k-medians in ℓ_1 and k-means.

1.1 Our Results

In this thesis, we provide approximation algorithms for explainable clustering with k-medians in ℓ_1, k-medians in ℓ_2, and k-means objectives.

We give a tight upper bound on the price of explainability for k-medians in ℓ_1. Specifically, we provide an efficient algorithm that transforms any clustering to an explainable clustering with

	k-medians in ℓ_1	k-medians in ℓ_2	k-means
Upper Bound	$O(\log k)$	$O(\log^{3/2} k)$	$O(k \log k)^{\dagger}$
Lower Bound	$\Omega(\log k)^{(*)}$	$\Omega(\log k)$	$\Omega(k/\log k)^{\S}$

Table 1.1: Summary of our results. The table shows known upper and lower bounds on the price of explainability for k-medians in ℓ_1 and ℓ_2, and for k-means. $(*)$: This lower bound for explainable k-medians is given by Dasgupta et al. (2020). $(\dagger)$: The upper bound for explainable k-means is improved to $O(k \log\log k)$ by Gupta et al. (2023). $(\S)$: The lower bound for explainable k-means is improved to $\Omega(k)$ by Esfandiari et al. (2022).

expected k-medians in ℓ_1 cost at most $2 \ln k + 2$ times the original k-medians in ℓ_1 cost. Note that we get an exponential improvement over the upper bound for the k-medians in ℓ_1 objective by Dasgupta et al. (2020). By adding a preprocessing step that embeds ℓ_2^2 into ℓ_1, we show that this algorithm also achieves an almost tight $O(k \log k)$ competitive ratio for explainable k-means. Furthermore, we present an algorithm for explainable k-medians in ℓ_2 with the competitive ratio bounded by $O(\log^{3/2} k)$. We complement these results with an almost tight lower bound of $\Omega(k/\log k)$ on the price of explainability for k-means and an $\Omega(\log k)$ lower bound on the price of explainability for k-medians in ℓ_2 objective. We summarise our results in Table 1.1.

Note that we improved the competitive ratio for explainable k-means to a near-optimal[1] bound of $\tilde{O}(k)$. This guarantee does not depend on the size and dimension of the data set. However, it is large for large data sets. For comparison, the competitive ratio for explainable k-medians is exponentially better than $\tilde{O}(k)$. It equals $O(\log k)$. Nevertheless, Dasgupta et al. (2020) and then Frost, Moshkovitz, and Rashtchian (2020) empirically demonstrated that, in practice, the price of explainability for k-means clustering is fairly small. In this work, we provide a theoretical justification for this observation. Specifically, we show a bi-criteria approximation algorithm which finds a threshold decision tree with $(1+\delta)k$ leaves and has a competitive ratio of $O(1/\delta \log^2 k \log\log k)$, where δ is a parameter between 0 and 1.

[1] It is possible to get a better competitive ratio for low dimensional data. For details, see Section 1.2

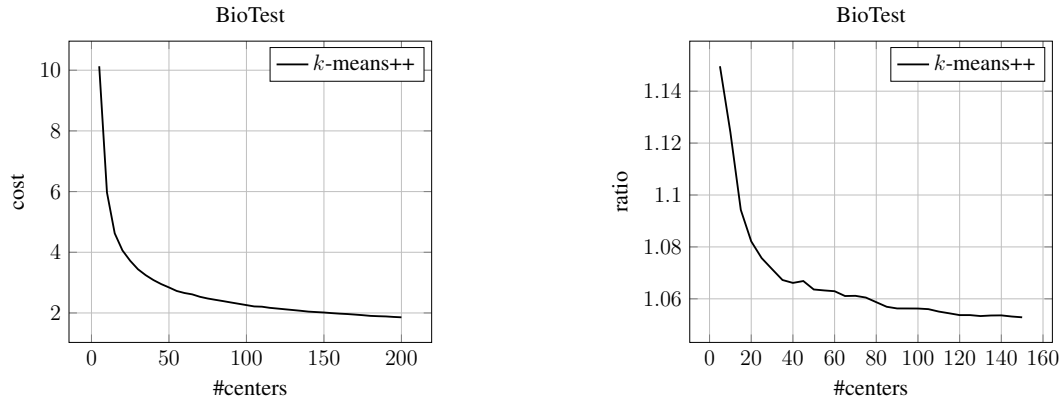


Figure 1.2: Performance of k-means++ on BioTest data set. The left diagram shows the cost of k-means++ for $k = 5, 10, 15, \ldots, 200$. The clustering cost is divided by the cost of k-means with 1000 clusters. The right diagram shows the ratio between the clustering cost with k centers and the cost with $(1+\delta)k$ centers for $k = 5, 10, \ldots, 150$ and $\delta = 0.2$.

We note that in practice the cost of the optimal k-means clustering is approximately the same for k and $(1+\delta)k$ clusters (here $\delta \in (0,1)$ is a small constant). In other words, for many data sets X, we have $\mathrm{OPT}_k(X) \approx \mathrm{OPT}_{(1+\delta)k}(X)$, where $\mathrm{OPT}_k(X)$ is the cost of the optimal unconstrained k-means clustering of X with k clusters[2]. The plot in Figure 1.2 shows that the cost of k-means++ clustering for BioTest data set from KDD Cup (Elber, 2004) is about the same for k and $(1+\delta)k$ centers when k is between 50 and 200. If $\mathrm{OPT}_k(X) \approx \mathrm{OPT}_{(1+\delta)k}(X)$, then our algorithm gives a *true* $\tilde{O}(\log^2 k)$ approximation, because

$$\mathrm{cost}_{\ell_2^2}(X, \mathcal{T}) \leq \tilde{O}(\log^2 k)\,\mathrm{OPT}_k(X) \approx \tilde{O}(\log^2 k)\,\mathrm{OPT}_{(1+\delta)k}(X).$$

We now formally state our results. We provide a randomized algorithm for finding bi-criteria explainable k-means. Similarly to the algorithm by Frost et al. (2020), our algorithm takes k centers $c^1, c^2, \ldots, c^k$ and a parameter $\delta > 0$ and returns a threshold decision tree $\mathcal{T}$ with $(1+\delta)k$

[2]In the worst case, we may have $\mathrm{OPT}_{(1+\delta)k}(X) \ll \mathrm{OPT}_k(X)$. For example, if X contains exactly $(1+\delta)k$ points, then $\mathrm{OPT}_{(1+\delta)k}(X) = 0$ but $\mathrm{OPT}_k(X) > 0$.

leaves. Each leaf of the tree is labeled with one of the centers $c^1, c^2, \ldots, c^k$. Let us denote the center returned by the decision tree $\mathcal{T}$ for point x by $\mathcal{T}(x)$. Then, the cost of explainable clustering defined by $\mathcal{T}$ equals

$$\mathrm{cost}_{\ell_2^2}(X, \mathcal{T}) \equiv \sum_{x \in X} \|x - \mathcal{T}(x)\|_2^2. \tag{1.7}$$

We show that there exists a polynomial-time randomized algorithm that given a data set X, a set of k centers $C = \{c^1, c^2, \ldots, c^k\}$, and parameter $\delta \in (0, 1)$, creates a threshold decision tree $\mathcal{T}$ whose leaves are labeled with centers from C. The expected number of leaves in $\mathcal{T}$ is $(1 + \delta)k$, and the expected cost of explainable clustering defined by $\mathcal{T}$ is

$$\mathbb{E}[\mathrm{cost}_{\ell_2^2}(X, \mathcal{T})] \leq O(1/\delta \cdot \log^2 k \log\log k) \cdot \mathrm{cost}(X, C).$$

Observe that our algorithm constructs a tree with $(1 + \delta)k$ leaves and only k centers. Thus, we can use this algorithm to partition X into k clusters. In this case, one cluster may be assigned to several different leaves. Alternatively, we can assign its own cluster to every leaf. Then, we will have a proper threshold decision tree with $(1 + \delta)k$ clusters. In either case, we can further improve the clustering by replacing the original center c^i assigned to each leaf u with the optimal center for the cluster assigned to u (the optimal center is the centroid of that cluster).

If C is the optimal set of centers for k means, then the explainable clustering provided by our algorithm has an expected cost of at most $O(1/\delta \cdot \log^2 k \log\log k)\,\mathrm{OPT}_k(X)$. Furthermore, if C is obtained by a constant factor bi-criteria approximation algorithm such as k-means++ (in which case, $|C| = (1 + \delta)k$ and $\mathrm{cost}(X, C) \leq O(1) \cdot \mathrm{OPT}_k(X)$), then the expected cost of the explainable clustering is also at most $O(1/\delta \cdot \log^2 k \log\log k)\,\mathrm{OPT}_k(X)$ and the number of leaves in the threshold decision tree is at most $(1 + 3\delta)k$ in expectation.

As we noted above, our work is influenced by the paper of Frost et al. (2020), who showed

a bi-criteria algorithm for explainable k-means. However, our algorithm for this problem is very different from theirs. It uses the approach from our previous paper (Makarychev and Shan (2021)). In that paper, we gave an algorithm for finding explainable k-*medians with* ℓ_2 *norm*. Our new algorithm has an additional crucial step: It duplicates some centers when the algorithm splits nodes. This step gives an exponential improvement to the competitive ratio for k-means. The analysis of our algorithm is considerably more involved than the analysis of the previous algorithm.

We complement our algorithmic results with an almost matching lower bound of $\Omega(1/\delta \cdot \log^2 k)$ for all threshold trees with at most $(1+\delta)k$ leaves. In Section B.1.4, we provide a family of k-means instances for which the greedy bi-criteria algorithm in Frost et al. (2020) finds a threshold tree $\mathcal{T}$ with $5k/4$ leaves of cost $\mathrm{cost}_{\ell_2^2}(X, \mathcal{T}) \geq \tilde{\Omega}(k^2)\,\mathrm{OPT}_k(X)$ for $k \to \infty$.

1.2 Related Work

Decision trees have been widely used for classification and clustering due to their simplicity. Examples of decision tree algorithms for supervised classification include CART by Breiman, Friedman, Olshen, and Stone (2017), ID3 by Quinlan (1986), and C4.5 by Quinlan (1993). Examples of decision tree algorithms for unsupervised clustering include algorithms by Liu, Xia, and Yu (2005), Fraiman, Ghattas, and Svarc (2013), Bertsimas, Orfanoudaki, and Wiberg (2018), and Saisubramanian, Galhotra, and Zilberstein (2020).

Dasgupta et al. (2020) proposed the problems of explainable k-medians in ℓ_1 and k-means. They defined these problems and offered algorithms for explainable k-means and k-medians with the competitive ratios of $O(k^2)$ and $O(k)$, respectively. Laber and Murtinho (2021), Makarychev and Shan (2021), Charikar and Hu (2022), Esfandiari, Mirrokni, and Narayanan (2022), and Gamlath, Jia, Polak, and Svensson (2021) provided improved upper and lower bounds on the price of explainability for k-means and k-medians. Particularly, Makarychev and Shan (2021), Esfandiari

et al. (2022), and Gamlath et al. (2021) gave an $\tilde{O}(k)$ competitive ratio for explainable k-means; and Makarychev and Shan (2021) and Esfandiari et al. (2022) gave an $\tilde{O}(\log k)$ competitive ratio for k-medians.

Laber and Murtinho (2021) gave $O(d \log k)$ and $O(dk \log k)$ competitive algorithms for explainable k-medians and k-means, respectively. They also provided $O(\sqrt{d}k^{1-1/d})$ upper bound and $\tilde{\Omega}(\sqrt{d}k^{1-1/d})$ lower bound for explainable k-center. They showed that the price of explainability for maximum-spacing clustering is $\Theta(n-k)$. Makarychev and Shan (2021) improved the competitive ratio to $O(\log k \log\log k)$ for explainable k-medians and $O(k \log k \log\log k)$ for explainable k-means. We also showed an $\Omega(k/\log k)$ lower bound on the price of explainability for k-means. Additionally, we gave an $\tilde{O}(\log^{3/2} n)$ competitive algorithm for explainable k-medians in ℓ_2 and an $\Omega(\log k)$ lower bound on the price of explainability for k-medians in ℓ_2. (The cost of a point x is $\mathrm{cost}_{\ell_2}(x,c) = \|x-c\|_2$.) Gamlath, Jia, Polak, and Svensson (2021) gave $O(\log^2 k)$ and $O(k \log^2 k)$ competitive algorithms for k-medians and k-means, respectively. They also provided an $O(k^{p-1}\log^2 k)$ competitive algorithm for explainable clustering with ℓ_p-norm objective for any $p \geq 1$. (The cost of a point x is $\mathrm{cost}_{\ell_p^p}(x,c) = \|x-c\|_p^p$.) Esfandiari et al. (2022) provided an $O(\log k \log\log k)$ competitive algorithm for explainable k-medians and an $O(k \log k)$ competitive algorithm for explainable k-means. They gave an $\Omega(k)$ lower bound for explainable k-means, which is slightly better than ours. They also gave an upper bound of $O(d \log^2 d)$ on the competitive ratio for explainable k-medians. This bound is better than $O(\log k)$ for small $d \ll \log k/\log\log k$. Charikar and Hu (2022) provided an algorithm that achieves $k^{1-2/d} \cdot \mathrm{poly}(d \log k)$ competitive ratio for explainable k-means (this algorithm gives stronger approximation guarantees when the dimension of the space, d, is small. For small $d \ll \log k/\log\log k$, their bound is better than $O(k)$.) They showed an almost matching $\Omega(k^{1-2/d}/\mathrm{ploy}\log k)$ lower bound for explainable k-means.

Boutsidis, Mahoney, and Drineas (2009), Boutsidis, Zouzias, Mahoney, and Drineas (2014), Cohen, Elder, Musco, Musco, and Persu (2015), Makarychev, Makarychev, and Razenshteyn (2019) and Becchetti, Bury, Cohen-Addad, Grandoni, and Schwiegelshohn (2019) showed how to reduce the dimensionality of a data set for k-means clustering. Particularly, Makarychev et al. (2019) proved that we can use the Johnson–Lindenstrauss transform to reduce the dimensionality of k-means to $d' = O(\log k)$. Note, however, that the Johnson–Lindenstrauss transform cannot be used for the explainable k-means, because this transform does not preserve the set of features. Instead, one can use a *feature selection* algorithm by Boutsidis et al. (2014) or Cohen et al. (2015) to reduce the dimensionality to $d' = \tilde{O}(k)$.

The algorithms for explainable k-medians by Makarychev and Shan (2021); Esfandiari, Mirrokni, and Narayanan (2022); Gamlath, Jia, Polak, and Svensson (2021) are variants of the same simple algorithm, which we call RANDOMCOORDINATECUT. Recently, Makarychev and Shan (2023) showed that the RANDOMCOORDINATECUT algorithm achieves $O(\log k)$ competitive ratio for k-medians, which matches the $\Omega(\log k)$ lower bound given by Dasgupta et al. (2020). Independently and concurrently with our work, Gupta, Pittu, Svensson, and Yuan (2023) proved a $O(\log k)$ upper bound on the price of explainability for k-medians. They showed that the competitive ratio of RANDOMCOORDINATECUT is $1 + H_{k-1}$, where H_k is the k-th harmonic number. Their work answers the open question raised by Gamlath, Jia, Polak, and Svensson (2021). They also proved a hardness of approximation result for explainable k-medians clustering and improved the competitive ratio for explainable k-means from $O(k \log k)$ to $O(k \log \log k)$.

Frost, Moshkovitz, and Rashtchian (2020) first considered the explainable clustering described by a threshold tree with more than k leaves. They provided some empirical evidence that bi-criteria algorithms for explainable k-means (that partition the data set into $(1 + \delta)k$ clusters) can give a much better competitive ratio than $O(k)$. Then, Makarychev and Shan (2022) gave a $\tilde{O}(\frac{1}{\delta} \log^2 k)$

competitive bi-criteria algorithm for explainable k-means by using a threshold tree with $(1+\delta)k$ leaves. We also provided an $\Omega(\frac{1}{\delta}\log^2 k)$ lower bound on competitive ratio for explainable k-means with $(1+\delta)k$ leaves.

Bandyapadhyay, Fomin, Golovach, Lochet, Purohit, and Simonov (2022) provided an algorithm that computes the optimal explainable k-medians and k-means clustering in time $n^{2d+O(1)}$ and $(4nd)^{k+O(1)}$, respectively. They also showed that it is NP-hard to find the optimal explainable k-medians and k-means clustering. Laber (2022) showed that it is NP-hard to approximate the optimal explainable clustering within $(1+\varepsilon)$ for some constant ε. Gupta et al. (2023) showed that the explainable k-medians and k-means can not be approximated within a factor of $O(\log k)$ unless P=NP.

Laber, Murtinho, and Oliveira (2023) proposed to use shallow decision trees for explainable clustering. They provided a heuristic algorithm that achieves comparable clustering costs with shallower threshold trees to previous algorithms by Dasgupta et al. (2020); Frost et al. (2020); Laber and Murtinho (2021) in experiments. Deng, Gavva, Patel, Karthik C. S., and Srinivasan (2023) showed the impossibility of depth reduction for explainable k-medians and k-means clustering. They found an instance in two-dimensional space $\mathbb{R}^2$ for which a threshold tree with depth $k-1$ has the same cost as the optimal unconstrained clustering, while any threshold tree with depth $k-2$ has an unbounded cost. Papanikolaou (2023) considered the explainable clustering on well-clusterable instances. He showed that if the instance is a-separated for some $a \geq 12kd^{1/p}$, then the greedy algorithm by Dasgupta et al. (2020) achieves a constant competitive ratio for explainable clustering with ℓ_p-norm. (The cost is $\text{cost}_{\ell_p^p}(x,c) = \|x-c\|_p^p$.) He also showed that the greedy algorithm achieves a constant competitive ratio for explainable k-medians in ℓ_1 if the instance is $\Omega(d)$-separated or the instance is $\Omega(\sqrt{d})$- perturbation stable.

The classic k-means and k-medians clustering has been extensively studied by researchers in

machine learning and theoretical computer science. Lloyd's algorithm (Lloyd (1982)) is the most popular heuristic for k-means clustering. Arthur and Vassilvitskii (2007) proposed a randomized seeding algorithm called k-means++, which achieves an expected $O(\log k)$ approximation. Ahmadian, Norouzi-Fard, Svensson, and Ward (2019) designed a primal-dual algorithm with an approximation factor of 6.357. It was improved to 6.12903 by Grandoni, Ostrovsky, Rabani, Schulman, and Venkat (2022). Recently, Cohen-Addad, Esfandiari, Mirrokni, and Narayanan (2022) improved the approximation factor to 5.912. Dasgupta (2008) and Aloise, Deshpande, Hansen, and Popat (2009) showed that k-means problem is NP-hard. Awasthi et al. (2015) showed that it is also NP-hard to approximate the k-means objective within a factor of $(1+\varepsilon)$ for some positive constant ε (see also Lee, Schmidt, and Wright (2017)). The bi-criteria approximation for k-means has also been studied before. Aggarwal, Deshpande, and Kannan (2009) proved that k-means++ that picks $(1+\delta)k$ centers gives a constant factor bi-criteria approximation for some constant $\delta > 0$. Later, Wei (2016) and Makarychev, Reddy, and Shan (2020) gave improved bi-criteria approximation guarantees for k-means++. Makarychev, Makarychev, Sviridenko, and Ward (2016) designed local search and LP-based algorithms with better bi-criteria approximation guarantees.

Charikar, Guha, Tardos, and Shmoys (1999) gave the first constant factor approximation algorithm for the unconstrained k-medians clustering in general metric spaces. Li and Svensson (2013) provided a $1+\sqrt{3}+\varepsilon$ approximation algorithm. Byrka, Pensyl, Rybicki, Srinivasan, and Trinh (2017) improved the approximation factor to $2.675+\varepsilon$. Cohen-Addad et al. (2022) recently improved the approximation factor to 2.406 for Euclidean k-medians. Megiddo and Supowit (1984) showed that the k-medians in ℓ_1 problem is NP-hard. Cohen-Addad and Lee (2022) showed that it is also NP-hard to approximate k-medians in ℓ_1 within a factor of 1.06.

CHAPTER 2

k-MEANS CLUSTERING

k-means clustering is one of the most commonly encountered unsupervised learning problems. Given a set of n data points $X = \{x_1, x_2, \ldots, x_n\}$ in Euclidean space $\mathbb{R}^d$, our goal is to partition them into k clusters (each characterized by a center), such that the sum of squared distances of data points to their nearest center is minimized. Specifically, we want to find a set C of k centers in $\mathbb{R}^d$ to minimize the total cost of clustering

$$\mathrm{cost}_{\ell_2^2}(X, C) := \sum_{x \in X} \min_{c \in C} \|x - c\|_2^2.$$

The most popular heuristic for solving this problem is Lloyd's algorithm Lloyd (1982), often referred to simply as "the k-means algorithm". Lloyd's algorithm uses iterative improvements to find a locally optimal k-means clustering. The performance of Lloyd's algorithm crucially depends on the quality of the initial clustering, which is defined by the initial set of centers, called a *seed*. Arthur and Vassilvitskii (2007) and Ostrovsky, Rabani, Schulman, and Swamy (2006) developed an elegant randomized seeding algorithm, known as the k-means++ algorithm. It works by choosing the first center uniformly at random from the data set and then choosing the subsequent $k - 1$ centers by randomly sampling a single point in each round with the sampling probability of every point proportional to its current cost. That is, the probability of choosing any data point x is proportional to the squared distance to its closest already chosen center. This squared distance is often denoted by $D^2(x)$. Arthur and Vassilvitskii (2007) proved that the expected cost of the initial clustering obtained by k-means++ is at most $8(\ln k + 2)$ times the cost of the optimal clustering

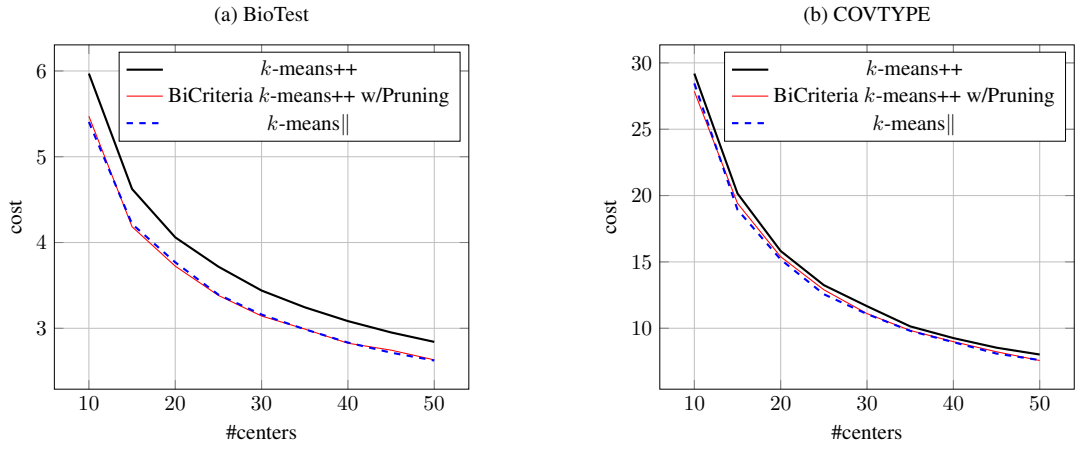


Figure 2.1: Performance of k-means++, k-means$\|$, and Bi-Criteria k-means++ with pruning on the BioTest and COVTYPE datasets. For $k = 10, 15, \cdots, 50$, we ran these algorithms for 50 iterations and took their average. We normalized the clustering costs. For each iteration, we divided the clustering costs by the cost given by k-means++ with 1000 centers.

i.e., k-means++ gives an $8(\ln k + 2)$-approximation for the k-means problem. They also provided a family of k-means instances for which the approximation factor of k-means++ is $2 \ln k$ and thus showed that their analysis of k-means++ is almost tight.

Due to its speed, simplicity, and good empirical performance, k-means++ is the most widely used algorithm for k-means clustering. It is employed by machine learning libraries such as Apache Spark MLlib, Google BigQuery, IBM SPSS, Intel DAAL, and Microsoft ML.NET. In addition to k-means++, these libraries implement a scalable variant of k-means++ called k-means$\|$ (read "k-means parallel") designed by Bahmani, Moseley, Vattani, Kumar, and Vassilvitskii (2012). Somewhat surprisingly, k-means$\|$ not only works better in parallel than k-means++ but also slightly outperforms k-means++ in practice in the single machine setting (see Bahmani et al. (2012) and Figure 2.1 below). However, theoretical guarantees for k-means$\|$ are substantially weaker than for k-means++.

The k-means$\|$ algorithm makes T passes over the data set (usually $T = 5$). In every round,

it independently draws approximately $\ell = \Theta(k)$ random centers according to the D^2 distribution. After each round, it recomputes the distances to the closest chosen centers and updates $D^2(x)$ for all data points in the data set. Thus, after T rounds, k-means$\|$ chooses approximately $T\ell$ centers. It then selects k centers among $T\ell$ centers using k-means++ on a weighted instance.

Our Contributions: In this section, we provide novel analyses of these two popular algorithms and show improved approximation and bi-criteria approximation guarantees for k-means++ and k-means$\|$. For any dataset $X \subseteq \mathbb{R}^d$ and any integer $k \geq 1$, we define the cost of the optimal solution for k-means problem to be

$$\mathrm{OPT}_k(X) := \min_{C,|C|=k} \mathrm{cost}(X, C) = \min_{C,|C|=k} \sum_{x \in X} \min_{c \in C} \|x - c\|_2^2.$$

We use $\mathrm{cost}_k(X) := \mathrm{cost}(X, C_k)$ to denote the cost of clustering given by k centers C_k sampled by k-means++. We use $\mathrm{cost}_T(X) := \mathrm{cost}(X, C_T)$ to denote the cost of clustering for centers C_T sampled by k-means$\|$ after T rounds.

We show that the expected cost of the solution output by k-means++ is at most $5(\ln k + 2)$ times the cost of the optimal solution,

$$\mathbb{E}[\mathrm{cost}_k(X)] \leq 5(\ln k + 2) \cdot \mathrm{OPT}_k(X).$$

This improves upon the bound of $8(\ln k + 2)$ shown by Arthur and Vassilvitskii (2007) and directly improves the approximation factors for several algorithms which use k-means++ as a subroutine like Local Search k-means++ (Lattanzi and Sohler, 2019).

Then, we address the question of why the observed performance of k-means$\|$ is better than the performance of k-means++. There are two possible explanations for this fact. (1) This may be the case because k-means$\|$ picks k centers in two stages. At the first stage, it samples $\ell T \geq k$ centers.

At the second stage, it prunes centers and chooses k centers among ℓT centers using k-means++. (2) This may also be the case because k-means$\|$ updates the distribution function $D^2(x)$ once in every round. That is, it recomputes $D^2(x)$ once for every ℓ chosen centers, while k-means++ recomputes $D^2(x)$ every time it chooses a center. In this paper, we empirically demonstrate that the first explanation is correct. First, we noticed that k-means$\|$ for $\ell \cdot T = k$ is almost identical with k-means++ (see Appendix A.1). Second, we compare k-means$\|$ with another algorithm called Bi-Criteria k-means++ with Pruning. This algorithm also works in two stages: At the Bi-Criteria k-means++ stage, it chooses $k + \Delta$ centers in the data set using k-means++. Then, at the Pruning stage, it picks k centers among the $k + \Delta$ centers selected at the first stage again using k-means++. Our experiments on the standard datasets BioTest from KDD-Cup 2004 Elber (2004) and COVTYPE from the UCI ML repository Dua and Graff (2017) show that the performance of k-means$\|$ and Bi-Criteria k-means++ with Pruning are essentially identical (see Figures 2.1 and Appendix A.1).

These results lead to another interesting question: How good are k-means++ and k-means$\|$ algorithms that sample $k+\Delta$ instead of k centers? The idea of oversampling using k-means++ was studied earlier in the literature under the name of *bi-criteria approximation*. Aggarwal, Deshpande, and Kannan (2009) showed that with constant probability, sampling $k + \Delta$ centers by k-means++ provides a constant-factor approximation if $\Delta \geq \delta k$ for some constant $\delta > 0$. Wei (2016) improved on this result by showing an expected approximation ratio of $8(1 + 1.618 \cdot k/\Delta)$. Note that for bi-criteria approximation, we compare the expected cost of the clustering with $k + \Delta$ centers they produce and the cost of the optimal clustering with exactly k centers $\mathrm{OPT}_k(X)$.

In this paper, we show that the expected bi-criteria approximation ratio for k-means++ with Δ

additional centers is at most the minimum of two bounds:

$$\text{(A) } 5\left(2+\frac{1}{2e}+\ln\frac{2k}{\Delta}\right) \text{ for } 1\le\Delta\le 2k; \text{ and (B) } 5\left(1+\frac{k}{e(\Delta-1)}\right) \text{ for } \Delta\ge 1.$$

Both bounds are better than the bound by Wei (2016). The improvement is especially noticeable for small values of Δ. More specifically, when the number of additional centers is $\Delta = k/\log k$, our approximation guarantee is $O(\log\log k)$ while Wei (2016) gives an $O(\log k)$ approximation.

We believe that our results for small values of Δ provide an additional explanation for why k-means++ works so well in practice. Consider a data scientist who wants to cluster a data set X with k^* *true clusters* (i.e. k^* latent groups). Since she does not know the actual value of k^*, she uses the *elbow method* (Boehmke and Greenwell, 2019) or some other heuristic to find k. Our results indicate that if she chooses slightly more number of clusters (for instance, $1.05k^*$), then she will get a constant bi-criteria approximation to the optimal clustering.

We also note that our bounds on the approximation factor smoothly transition from the regular ($\Delta = 0$) to bi-criteria ($\Delta > 0$) regime. We complement our analysis with an almost matching lower bound of $\Theta(\log(k/\Delta))$ on the approximation factor of k-means for $\Delta \le k$ (see Appendix A.2).

We then analyze Bi-Criteria k-means$\|$ algorithm, the variant of k-means$\|$ that does not prune centers at the second stage. In their original paper, Bahmani, Moseley, Vattani, Kumar, and Vassilvitskii (2012) showed that the expected cost of the solution for k-means$\|$ with T rounds and oversampling parameter ℓ is at most:

$$\frac{16}{1-\alpha}\mathrm{OPT}_k(X) + \left(\frac{1+\alpha}{2}\right)^T \mathrm{OPT}_1(X),$$

where $\alpha = \exp(-(1-e^{-\ell/(2k)}))$; $\mathrm{OPT}_k(X)$ is the cost of the optimal k-means clustering of X; $\mathrm{OPT}_1(X)$ is the cost of the optimal clustering of X with one center. We note that $\mathrm{OPT}_1(X) \gg$

$\mathrm{OPT}_k(X)$. For $\ell = k$, this result gives a bound of $\approx 49\,\mathrm{OPT}_k(X) + 0.83^T\mathrm{OPT}_1(X)$. Bachem, Lucic, and Krause (2017) improved the approximation guarantee for $\ell \geq k$ to

$$26\mathrm{OPT}_k(X) + 2\Big(\frac{k}{e\ell}\Big)^T\mathrm{OPT}_1(X).$$

In this work, we improve this bound for $\ell \geq k$ and also obtain a better bound for $\ell < k$. For $\ell \geq k$, we show that the cost of k-means$\|$ without pruning is at most

$$8\mathrm{OPT}_k(X) + 2\Big(\frac{k}{e\ell}\Big)^T\mathrm{OPT}_1(X).$$

For $\ell < k$, we give a bound of

$$\frac{5}{1-e^{-\frac{\ell}{k}}}\,\mathrm{OPT}_k(X) + 2\left(e^{-\frac{\ell}{k}}\right)^T\mathrm{OPT}_1(X)$$

Organization: We first describe a general framework for analyzing k-means++ and k-means$\|$ algorithms in section 2.1. Then, we show an improved $5(\ln k + 2)$ approximation for k-means++ in section 2.2. In section 2.3, we provide better guarantees for bi-criteria k-means++. Finally, in section 2.4, we give a better analysis for k-means$\|$.

2.1 General framework

In this section, we describe a general framework we use to analyze k-means++ and k-means$\|$. First, we formally describe k-means++ and k-means$\|$ algorithms. We also introduce a different implementation of k-means$\|$, called k-means$\|_{\mathrm{Pois}}$.

k-means++ seeding: The k-means++ algorithm samples the first center uniformly at random from the given points and then samples $k-1$ centers sequentially from the given points with the

probability of each point being sampled proportional to its cost i.e. $\text{cost}(x, C)/\text{cost}(X, C)$. See Algorithm 1.

Algorithm 1 k-means++ seeding

1: Sample a point c uniformly at random from X and set $C_1 = \{c\}$.
2: **for** $t = 2$ **to** k **do**
3: Sample $x \in X$ w.p. $\text{cost}(x, C_{t-1})/\text{cost}(X, C_{t-1})$.
4: $C_t = C_{t-1} \cup \{x\}$.
5: **end for**
6: **Return** C_k

k-means$\|$ and k-means$\|_{\text{Pois}}$ seeding: In the k-means$\|$ algorithm, the first center is chosen uniformly at random from X. But after that, at each round, the algorithm samples each point independently with probability $\min\{\ell \cdot \text{cost}(x, C)/\text{cost}(X, C), 1\}$ where ℓ is the *oversampling parameter* chosen by the user and it usually lies between $0.1k$ and $10k$. The algorithm runs for T rounds (where T is also a parameter chosen by the user) and samples around ℓT points, which is usually strictly larger than k. This oversampled set is then weighted using the original data set X and a weighted version of k-means++ is run on this set to get the final k-centers. We only focus on the stage in which we get the oversampled set because the guarantees for the second stage come directly from k-means++. The k-means$\|$ seeding is shown in Algorithm 2.

For the sake of analysis, we also consider a different implementation of k-means$\|$, which we call k-means$\|_{\text{Pois}}$ (Algorithm 3). This algorithm differs from k-means$\|$ in that each point is sampled independently with probability $1 - \exp(-\ell \cdot \text{cost}(x, C)/\text{cost}(X, C))$ rather than $\min\{\ell \cdot \text{cost}(x, C)/\text{cost}(X, C), 1\}$. In practice, there is essentially no difference between k-means$\|$ and k-means$\|_{\text{Pois}}$, since $\ell \cdot \text{cost}(x, C)/\text{cost}(X, C)$ is a very small number for all x and thus the sampling probabilities for k-means$\|$ and k-means$\|_{\text{Pois}}$ are almost equal.

We then give a general framework for analyzing k-means++ and k-means$\|_{\text{Pois}}$ algorithm. Let C_t be the set of centers chosen by this algorithm after step t. For the sake of analysis, we assume

Algorithm 2 k-means$\|$ seeding

1: Sample a point c uniformly from X and set $C_1 = \{c\}$.
2: **for** $t = 1$ **to** T **do**
3: Sample each point x into C' independently w.p. $\min\{1, \lambda_t(x)\}$ where $\lambda_t(x) = \ell \cdot \mathrm{cost}(x, C_t)/\mathrm{cost}(X, C_t)$.
4: Let $C_{t+1} = C_t \cup C'$.
5: **end for**

Algorithm 3 k-means$\|_{\mathrm{Pois}}$ seeding

1: Sample a point c uniformly from X and set $C_1 = \{c\}$
2: **for** $t = 1$ **to** T **do**
3: Sample each point x into C' independently w.p. $1 - e^{-\lambda_t(x)}$ where $\lambda_t(x) = \ell \cdot \mathrm{cost}(x, C_t)/\mathrm{cost}(X, C_t)$
4: Let $C_{t+1} = C_t \cup C'$.
5: **end for**

that C_t is an ordered set or list of centers and the order of centers in C_t is the same as the order in which our algorithm chooses these centers. We explain how to order centers in k-means$\|_{\mathrm{Pois}}$ algorithm in Section 2.4. We denote by T the stopping time of the algorithm. Observe that after step t of the algorithm, the probabilities of choosing a new center in k-means++ or a batch of new centers in k-means$\|_{\mathrm{Pois}}$ are defined by the current costs of points in X which, in turn, are completely determined by the current set of centers C_t. Thus, the states of the algorithm form a Markov chain.

In our analysis, we fix the optimal clustering $\mathcal{P} = \{P_1, \ldots, P_k\}$ (if this clustering is not unique, we pick an arbitrary optimal clustering). The optimal cost of each cluster P_i is $\mathrm{OPT}_1(P_i)$ and the optimal cost of the entire clustering is $\mathrm{OPT}_k(X) = \sum_{i=1}^{k} \mathrm{OPT}_1(P_i)$.

Following the k-means++ paper by Arthur and Vassilvitskii (2007), we say that a cluster P_i is *hit* or *covered* by a set of centers C if $C \cap P_i \neq \varnothing$; otherwise, we say that P_i is *not hit* or *uncovered*. We split the cost of each cluster P_i into two components which we call the covered and uncovered

costs of P_i. For a given set of centers C,

$$\text{The covered or hit cost of } P_i, \qquad H(P_i, C) := \begin{cases} \text{cost}(P_i, C), & \text{if } P_i \text{ is covered by } C \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{The uncovered cost of } P_i, \qquad U(P_i, C) := \begin{cases} 0, & \text{if } P_i \text{ is covered by } C \\ \text{cost}(P_i, C), & \text{otherwise.} \end{cases}$$

Let $H(X, C) = \sum_{i=1}^{k} H(P_i, C)$ and $U(X, C) = \sum_{i=1}^{k} U(P_i, C)$. Then, the total cost of clustering is the sum of covered cost and uncovered cost,

$$\text{cost}(X, C) = H(X, C) + U(X, C).$$

For brevity, denote $\text{cost}_t(Y) = \text{cost}(Y, C_t)$ for any $Y \subseteq X$, $H_t(P_i) = H(P_i, C_t)$, and $U_t(P_i) = U(P_i, C_t)$. In Section 2.2, we show that for any t, we have $\mathbb{E}[H_t(X)] \leq 5\text{OPT}_k(X)$, which is an improvement over the bound of $8\text{OPT}_k(X)$ given by Arthur and Vassilvitskii (2007). Then, in Sections 2.3 and 2.4, we analyze the expected uncovered cost $U(X, C_T)$ for k-means++ and k-means$\|_{\text{Pois}}$ algorithms.

Consider a center c in C. We say that c is a *miss* if another center c' covers the same cluster $P_i \in \mathcal{P}$ as c, and c' appears before c in the ordered set C. We denote the number of misses in C by $M(C)$ and the number of clusters in $\mathcal{P}$ not covered by centers in C by $K(C)$.

Observe that the stochastic processes $U_t(P_i)$ with discrete time t are non-increasing since the algorithm never removes centers from the set C_t and therefore the distance from any point $x \in X$ to C_t never increases. Similarly, the processes $H_t(P_i)$ are non-increasing after step t_i when P_i is covered for the first time. In this paper, we sometimes use a proxy $\widetilde{H}_t(P_i)$ for $H_t(P_i)$, which we

define as follows. If P_i is covered by C_t, then $\widetilde{H}_t(P_i) = H_{t_i}(P_i)$, where $t_i \leq t$ is the first time when P_i is covered by C_t. If P_i is not covered by C_t, then $\widetilde{H}_t(P_i) = 5\mathrm{OPT}_1(P_i)$. It is easy to see that $H_t(P_i) \leq \widetilde{H}_{t'}(P_i)$ for all $t \leq t'$. In Section 2.2, we also show that $\widetilde{H}_t(P_i)$ is a supermartingale i.e., $\mathbb{E}[\widetilde{H}_{t'}(P_i) \mid C_t] \leq \widetilde{H}_t(P_i)$ for all $t \leq t'$.

2.2 Analysis of k-means++

We first analyze the popular k-means++ algorithm. We show that the expected cost of the solution output by k-means++ is at most $5(\ln k + 2)$ times the cost of the optimal solution. This improves upon the bound of $8(\ln k + 2)$ shown by Arthur and Vassilvitskii (2007).

Theorem 2.1. *The approximation factor of k-means++ is at most $5(\ln k + 2)$.*

The k-means++ algorithm samples the first center uniformly at random from the given points and then samples $k - 1$ centers sequentially from the given points with probability of each point being sampled proportional to its cost i.e. $\mathrm{cost}(x, C)/\mathrm{cost}(X, C)$.

We improve the bound by Arthur and Vassilvitskii (2007) on the expected cost of a covered cluster in k-means++. Pick an arbitrary cluster P_i in the optimal solution $\mathcal{P} = \{P_1, \ldots, P_k\}$ and consider an arbitrary state $C_t = \{c_1, \ldots, c_t\}$ of the k-means++ algorithm. Let c_{t+1} be the new center which the algorithm adds to C_t at step $t + 1$. Suppose now that the new center c_{t+1} cover P_i i.e. $c_{t+1} \in P_i$. We show that the expected cost of cluster P_i after step $t + 1$ conditioned on the event $\{c_{t+1} \in P_i\}$ and the current state of the algorithm C_t is upper bounded by $5\mathrm{OPT}_1(P_i)$ i.e.

$$\mathbb{E}[\mathrm{cost}(P_i, C_{t+1}) \mid C_t, c_{t+1} \in P_i] \leq 5\mathrm{OPT}_1(P_i). \tag{2.1}$$

We now prove the main lemma.

Lemma 2.2. *Consider an arbitrary set of centers* $C = \{c_1, \ldots, c_t\} \subseteq \mathbb{R}^d$ *and an arbitrary set* $P \subseteq X$. *Pick a random point* c *in* P *with probability* $\Pr\{c = x\} = \mathrm{cost}(x, C)/\mathrm{cost}(P, C)$. *Let* $C' = C \cup \{c\}$. *Then,* $\mathbb{E}_c[\mathrm{cost}(P, C')] \leq 5\mathrm{OPT}_1(P)$.

Remarks: Lemma 2.2 in the paper by Arthur and Vassilvitskii (2007) gives a bound of $8\mathrm{OPT}_1(P)$.

Proof. The cost of any point y after picking center c equals the squared distance from y to the set of centers $C' = C \cup \{c\}$, which in turn equals $\min\{\mathrm{cost}(y, C), \|y - c\|_2^2\}$. Thus, if a point $x \in P$ is chosen as a center, then the cost of point y equals $\min\{\mathrm{cost}(y, C), \|x - y\|_2^2\}$. Since $\Pr\{c = x\} = \mathrm{cost}(x, C)/\mathrm{cost}(P, C)$, we have

$$\mathbb{E}_c[\mathrm{cost}(P, C')] = \sum_{\substack{x \in P \\ y \in P}} \frac{\mathrm{cost}(x, C)}{\mathrm{cost}(P, C)} \cdot \min\{\mathrm{cost}(y, C), \|x - y\|_2^2\}.$$

We write the right hand side in a symmetric form with respect to x and y. To this end, we define a function f as follows:

$$f(x, y) = \mathrm{cost}(x, C) \cdot \min\{\|x - y\|_2^2, \mathrm{cost}(y, C)\} + \mathrm{cost}(y, C) \cdot \min\{\|x - y\|_2^2, \mathrm{cost}(x, C)\}.$$

Note that $f(x, y) = f(y, x)$. Then,

$$\mathbb{E}_c[\mathrm{cost}(P, C')] = \frac{1}{2\mathrm{cost}(P, C)} \sum_{(x,y) \in P \times P} f(x, y).$$

We now give an upper bound on $f(x, y)$ and then use this bound to finish the proof of Lemma 2.2.

Lemma 2.3. *For any* $x, y \in P$, *we have* $f(x, y) \leq 5 \cdot \min\{\mathrm{cost}(x, C), \mathrm{cost}(y, C)\} \cdot \|x - y\|_2^2$.

Proof. Since $f(x, y)$ is a symmetric function with respect to x and y, we may assume without loss

of generality that $\text{cost}(x, C) \leq \text{cost}(y, C)$. Then, we need to show that $f(x, y) \leq 5\text{cost}(x, C) \cdot \|x - y\|_2^2$. Consider the following three cases.

Case 1: If $\text{cost}(x, C) \leq \text{cost}(y, C) \leq \|x - y\|_2^2$, then

$$f(x, y) = 2\text{cost}(x, C) \cdot \text{cost}(y, C) \leq 2\text{cost}(x, C) \cdot \|x - y\|_2^2.$$

Case 2: If $\text{cost}(x, C) \leq \|x - y\|_2^2 \leq \text{cost}(y, C)$, then

$$f(x, y) = \text{cost}(x, C) \cdot \|x - y\|_2^2 + \text{cost}(y, C) \cdot \text{cost}(x, C).$$

By the triangle inequality, we have

$$\text{cost}(y, C) \leq \{\sqrt{\text{cost}(x, C)} + \|x - y\|_2\}^2 \leq 4\|x - y\|_2^2.$$

Thus, $f(x, y) \leq 5\text{cost}(x, C) \cdot \|x - y\|_2^2$.

Case 3: If $\|x - y\|_2^2 \leq \text{cost}(x, C) \leq \text{cost}(y, C)$, then

$$f(x, y) = (\text{cost}(x, C) + \text{cost}(y, C))\|x - y\|_2^2.$$

By the triangle inequality,

$$\text{cost}(y, C) \leq \{\sqrt{\text{cost}(x, C)} + \|x - y\|_2\}^2 \leq 4\text{cost}(x, C).$$

Thus, we have $f(x, y) \leq 5\text{cost}(x, C) \cdot \|x - y\|_2^2$.

In all cases, the desired inequality holds. This concludes the proof of Lemma 2.3. □

We use Lemma 2.3 to bound the expected cost of P. Let ϕ^* be a vector in $\mathbb{R}^P$ with $\phi^*_x = \text{cost}(x, C)$ for any $x \in P$. Then, by Lemma 2.3, $f(x, y) \leq 5 \min\{\phi^*_x, \phi^*_y\}\|x - y\|_2^2$. Since $\text{cost}(P, C) = \sum_{z \in P} \phi^*_z$, we have

$$\mathbb{E}_c[\text{cost}(P, C')] \leq \underbrace{\frac{5 \sum_{(x,y) \in P \times P} \min\{\phi^*_x, \phi^*_y\}\|x - y\|_2^2}{2 \sum_{z \in P} \phi^*_z}}_{5F(\phi^*)}.$$

For arbitrary vector $\phi \in \mathbb{R}^P_{\geq 0}$, define the following function:

$$F(\phi) = \frac{\sum_{(x,y) \in P \times P} \min\{\phi_x, \phi_y\}\|x - y\|_2^2}{2 \sum_{z \in P} \phi_z}. \tag{2.2}$$

We have $\mathbb{E}_c[\text{cost}(P, C')] \leq 5F(\phi^*)$. Thus, to finish the proof of Lemma 2.2, it suffices to show that $F(\phi) \leq \text{OPT}_1(P)$ for every $\phi \geq 0$ and particularly for $\phi = \phi^*$. By Lemma 2.4 (which we state and prove below), the function $F(\phi)$ is maximized when $\phi \in \{0, 1\}^P$. Let ϕ^{**} be a maximizer of $F(\phi)$ in $\{0, 1\}^P$ and $P' = \{x \in P : \phi^{**}_x = 1\}$. Observe that

$$F(\phi^{**}) = \frac{\sum_{(x,y) \in P' \times P'} \|x - y\|_2^2}{2|P'|} = \text{OPT}_1(P').$$

Here we used the closed form expression for the optimal cost of cluster P'

$$\text{OPT}_1(P') = \sum_{x \in P'} \|x - \mu(P')\|^2 = \frac{\sum_{(x,y) \in P' \times P'} \|x - y\|^2}{2|P'|},$$

where $\mu(P')$ is the mean of all points in P'. Since $P' \subset P$, we have $\text{OPT}_1(P') \leq \text{OPT}_1(P)$. Thus, $F(\phi^*) \leq F(\phi^{**}) \leq \text{OPT}_1(P)$. □

Lemma 2.4. *There exists a maximizer ϕ^{**} of $F(\phi)$ in the region $\{\phi \geq 0\}$ such that $\phi \in \{0, 1\}^P$.*

Proof. Let $m = |P|$ be the size of the cluster P and Π be the set of all bisections or permutations $\pi : \{1, \ldots, m\} \to P$. Partition the set $\{\phi \geq 0\}$ into $m!$ regions ("cones over order polytopes"):

$$\{\phi : \phi \geq 0\} = \cup_{\pi \in \Pi} O_\pi,$$

where $O_\pi = \{\phi : 0 \leq \phi_{\pi(1)} \leq \phi_{\pi(2)} \leq \cdots \leq \phi_{\pi(m)}\}$. We show that for every $\pi \in \Pi$, there exists a maximizer ϕ^{**} of $F(\phi)$ in the region O_π, such that $\phi^{**} \in \{0,1\}^P$. Therefore, there exists a global maximizer ϕ^{**} that belongs $\{0,1\}^P$

Fix a $\pi \in \Pi$. Denote by V the hyperplane $\{\phi : \sum_{x \in P} \phi_x = 1\}$. Observe that F is a scale invariant function i.e., $F(\phi) = F(\lambda\phi)$ for every $\lambda > 0$. Thus, for every $\phi \in O_\pi$, there exists a $\phi' \in O_\pi \cap V$ (namely, $\phi' = \phi/(\sum_{x \in P} \phi_x)$) such that $F(\phi') = F(\phi)$. Hence, $\max\{F(\phi) : \phi \in O_\pi\} = \max\{F(\phi) : \phi \in O_\pi \cap V\}$. Note that for $\phi \in V$, the denominator of (2.2) equals 2, and for $\phi \in O_\pi$, the numerator of (2.2) is a linear function of ϕ. Therefore, $F(\phi)$ is a linear function in the convex set $O_\pi \cap V$. Consequently, one of the maximizers of F must be an extreme point of $O_\pi \cap V$.

The polytope $O_\pi \cap V$ is defined by m inequalities and one equality. Thus, for every extreme point ϕ of this polytope, all inequalities $\phi_{\pi(i)} \leq \phi_{\pi(i+1)}$ but one must be tight. In other words, for some $j < m$, we have

$$0 = \phi_{\pi(1)} = \cdots = \phi_{\pi(j)} < \phi_{\pi(j+1)} = \cdots = \phi_{\pi(m)}. \tag{2.3}$$

Therefore, there exists a maximizer ϕ of $F(\phi)$ in $O_\pi \cap V$ satisfying (2.3) for some j. After rescaling ϕ – multiplying all coordinates of ϕ by $(m - j)$ – we obtain a vector ϕ^{**} whose first j coordinates $\phi^{**}_{\pi(1)}, \ldots, \phi^{**}_{\pi(j)}$ are zeroes and the last $m - j$ coordinates $\phi^{**}_{\pi(j+1)}, \ldots, \phi^{**}_{\pi(m)}$ are ones. Thus, $\phi^{**} \in \{0,1\}^P$. Since F is rescaling invariant, $F(\phi^{**}) = F(\phi)$. This concludes the proof. □

Replacing the bound in Lemma 3.2 from the analysis of Arthur and Vassilvitskii (2007) by our bound from Lemma 2.2 gives Theorem 2.1.

We now state an important corollary of Lemma 2.2.

Corollary 2.5. *For every $P \in \mathcal{P}$, the process $\widetilde{H}_t(P)$ for k-means++ is a supermartingale i.e.,*

$$\mathbb{E}[\widetilde{H}_{t+1}(X) \mid C_t] \leq \widetilde{H}_t(X).$$

Proof. The value of $\widetilde{H}_t(X)$ changes only if at step t, we cover a yet uncovered cluster P. In this case, the value of $\widetilde{H}_{t+1}(P)$ changes by the new cost of P minus $5\mathrm{OPT}(P)$. By Lemma 2.2 this quantity is non-positive in expectation. □

Since the process $\widetilde{H}_t(P)$ is a supermartingale, we have $\mathbb{E}[\widetilde{H}_t(P)] \leq \widetilde{H}_0(P) = 5\mathrm{OPT}_1(P)$. Hence, $\mathbb{E}[H_t(P)] \leq \mathbb{E}[\widetilde{H}_t(P)] = 5\mathrm{OPT}_1(P)$. Thus, $\mathbb{E}[H_t(X)] \leq 5\mathrm{OPT}_k(X)$. Since $\mathrm{cost}_t(X) = H_t(X) + U_t(X)$ and we have a bound on the expectation of the covered cost, $H_t(X)$, in the remaining sections, we shall only analyze the uncovered cost $U_t(X)$.

2.3 Bi-criteria Approximation of k-means++

In this section, we give a bi-criteria approximation guarantee for k-means++.

Theorem 2.6. *Let $\mathrm{cost}_{k+\Delta}\{X\}$ be the cost of the clustering with $k + \Delta$ centers sampled by the k-means++ algorithm. Then, for $\Delta \geq 1$, the expected cost $\mathbb{E}\{\mathrm{cost}_{k+\Delta}(X)\}$ is upper bounded by (below $(a)^+$ denotes $\max(a, 0)$).*

$$\min\left\{2 + \frac{1}{2e} + \left(\ln\frac{2k}{\Delta}\right)^+, 1 + \frac{k}{e(\Delta - 1)}\right\} 5\mathrm{OPT}_k(X).$$

Note that the above approximation guarantee is the minimum of two bounds: (1) $2 + \frac{1}{2e} + \ln \frac{2k}{\Delta}$ for $1 \leq \Delta \leq 2k$; and (2) $1 + \frac{k}{e(\Delta-1)}$ for $\Delta \geq 1$. The second bound is stronger than the first bound when $\Delta/k \gtrapprox 0.085$.

We now present a high-level overview of the proof and then give a formal proof. Our proof consists of three steps.

First, we prove bound (2) on the expected cost of the clustering returned by k-means++ after $k + \Delta$ rounds. We argue that the expected cost of the covered clusters is bounded by $5\mathrm{OPT}_k(X)$ and thus it is sufficient to bound the expected cost of uncovered clusters. Consider an optimal cluster $P \in \mathcal{P}$. We need to estimate the probability that it is not covered after $k + \Delta$ rounds. We upper bound this probability by the probability that the algorithm does not cover P before it makes Δ misses (note: after $k + \Delta$ rounds k-means++ must make at least Δ misses).

In this overview, we make the following simplifying assumptions (which turn out to be satisfied in the worst case for bi-criteria k-means++): Suppose that the uncovered cost of cluster P does not decrease before it is covered and equals $U(P)$ and, moreover, the total cost of all covered clusters almost does not change and equals $H(X)$ (this may be the case if one large cluster contributes most of the covered cost, and that cluster is covered at the first step of k-means++). Under these assumptions, the probability that k-means++ chooses Δ centers in the already covered clusters and does not choose a single center in P equals $(H(X)/(U(P) + H(X)))^{\Delta}$. If k-means++ does not choose a center in P, the *uncovered* cost of cluster P is $U(P)$; otherwise, the *uncovered* cost of cluster P is 0. Thus, the expected *uncovered cost* of P is $(H(X)/(U(P) + H(X)))^{\Delta}U(P)$. It is easy to show that $(H(X)/(U(P) + H(X)))^{\Delta}U(P) \leq H(X)/(e(\Delta - 1))$. Thus, the expected *uncovered cost* of all clusters is at most

$$\frac{k}{(e(\Delta - 1))}\mathbb{E}[H(X)] \leq \frac{k}{(e(\Delta - 1))}5\mathrm{OPT}_k(X).$$

Then, we use ideas from Arthur and Vassilvitskii (2007), Dasgupta (2013) to prove the following statement: Let us count the cost of uncovered clusters only when the number of misses after k rounds of k-means++ is greater than $\Delta/2$. Then the expected cost of uncovered clusters is at most $O(\log(k/\Delta)) \cdot \mathrm{OPT}_k(X)$. That is, $\mathbb{E}[H(U_k(X) \cdot \mathbf{1}\{M(C_k) \geq \Delta/2\}] \leq O(\log(k/\Delta)) \cdot \mathrm{OPT}_k(X)$.

Finally, we combine the previous two steps to get bound (1). We argue that if the number of misses after k rounds of k-means++ is less than $\Delta/2$, then almost all clusters are covered. Hence, we can apply bound (2) to $k' \leq \Delta/2$ uncovered clusters and Δ remaining rounds of k-means++ and get a $5(1+1/(2e))$ approximation. If the number of misses is greater than $\Delta/2$, then the result from the previous step yields an $O(\log(k/\Delta))$ approximation.

In this section, we analyze the bi-criteria k-means++ algorithm and prove Theorem 2.6. To this end, we establish the first and second bounds from Theorem 2.6 on the expected cost of the clustering after $k + \Delta$ rounds of k-means. We will start with the second bound.

2.3.1 Large Number of Extra Centers

Lemma 2.7. *The following bi-criteria bound holds*

$$\mathbb{E}[\mathrm{cost}_{k+\Delta}(X)] \leq 5\left(1 + \frac{k}{e(\Delta - 1)}\right)\mathrm{OPT}_k(X).$$

Consider the discrete time Markov chain C_t associated with k-means++ algorithm (see Section 2.1). Let $P \in \mathcal{P}$ be an arbitrary cluster in the optimal solution. Partition all states of the Markov chain into $k + \Delta$ disjoint groups $\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_{k+\Delta-1}$ and $\mathcal{H}$. Each set $\mathcal{M}_i$ contains all states C with i misses that do not cover P: $\mathcal{M}_i = \{C : M(C) = i, P \cap C = \varnothing\}$, where $M(C)$ is the number of misses for centers C. The set $\mathcal{H}$ contains all states C that cover P: $\mathcal{H} = \{C : P \cap C \neq \varnothing\}$.

We now define a new Markov chain S_t. To this end, we first expand the set of states $\{C\}$. For every state C of the process C_t, we create two additional "virtual" states C^a and C^b. Then, we let $S_{2t} = C_t$ for every even step $2t$, and

$$S_{2t+1} = \begin{cases} C_t^a, & \text{if } C_t, C_{t+1} \in \mathcal{M}_i \\ C_t^b, & \text{if } C_t \in \mathcal{M}_i, C_{t+1} \in \mathcal{M}_{i+1} \cup \mathcal{H}. \end{cases}$$

for every odd step $2t+1$. We stop S_t when C_t stops or when C_t hits the set $\mathcal{H}$ (i.e., $C_t \in \mathcal{H}$). Loosely speaking, S_t follows Markov chain C_t but makes additional intermediate stops. When C_t moves from one state in $\mathcal{M}_i$ to another state in $\mathcal{M}_i$, S_{2t+1} stops in C_t^a; and when C_t moves from a state in $\mathcal{M}_i$ to a state in $\mathcal{M}_{i+1}$ or $\mathcal{H}$, S_{2t+1} stops in C_t^b.

Write transition probabilities for S_t:

$$\Pr(S_{2t+1} = C^a \mid S_{2t} = C) = \frac{U(X,C) - U(P,C)}{\text{cost}(X,C)},$$
$$\Pr(S_{2t+1} = C^b \mid S_{2t} = C) = \frac{U(P,C) + H(X,C)}{\text{cost}(X,C)},$$

and for all $C \in \mathcal{M}_i$ and $C' = C \cup \{x\} \in \mathcal{M}_i$,

$$\Pr(S_{2t+2} = C' \mid S_{2t+1} = C^a) = \frac{\text{cost}(x,C)}{U(X,C) - U(P,C)},$$

for all $C \in \mathcal{M}_i$ and $C' = C \cup \{x\} \in \mathcal{M}_{i+1} \cup \mathcal{H}$,

$$\Pr(S_{2t+2} = C' \mid S_{2t+1} = C^b) = \frac{\text{cost}(x,C)}{U(P,C) + H(X,C)}.$$

Above, $U(X,C) - U(P,C)$ is the cost of points in all uncovered clusters except for cluster P. If we

pick a center from these clusters, we will necessarily cover a new cluster, and therefore S_{2t+2} will stay in $\mathcal{M}_i$. Similarly, $U(P,C) + H(X,C)$ is the cost of all covered clusters plus the uncovered cost of P. If we pick a center from these clusters, then S_{2t+2} will move to $\mathcal{M}_{i+1}$ (if the center hits a covered cluster) or $\mathcal{H}$ (if the center hits cluster P).

Define another Markov chain Y_t. The transition probabilities of Y_t are the same as the transition probabilities of S_t except Y_t never visits states in $\mathcal{H}$ and therefore for $C \in \mathcal{M}_i$ and $C' = C \cup \{x\} \in \mathcal{M}_{i+1}$, we have

$$\Pr(Y_{2t+2} = C' \mid Y_{2t+1} = C^b) = \frac{\mathrm{cost}(x,C)}{H(X,C)}.$$

We now prove a lemma that relates probabilities of visiting states by S_t and Y_t.

Lemma 2.8. *For every $t \le k + \Delta$ and states $C' \in \mathcal{M}_i$, $C'' \in \mathcal{M}_\Delta$, we have*

$$\frac{\Pr(C'' \in \{S_j\} \mid S_{2t} = C')}{\Pr(C'' \in \{Y_j\} \mid Y_{2t} = C')} \le \left(\frac{\widetilde{H}(X,C'')}{\widetilde{H}(X,C'') + U(P,C'')} \right)^{\Delta - i}$$

where $\{C'' \in \{S_j\}\}$ and $\{C'' \in \{Y_j\}\}$ denote the events that Markov chain S_t visits C'' and Markov chain Y_t visits C'', respectively.

Proof. Consider the unique path p from C' to C'' in the state space of S (note that the transition graphs for S and Y are directed trees). The probability of transitioning from C' to C'' for S and Y equals the product of respective transition probabilities for every edge on the path. Recall that transition probabilities for S and Y are the same for all states but C^b, where $C \in \cup_j \mathcal{M}_j$. The number of such states on the path p is equal to the number transitions from $\mathcal{M}_j$ to $\mathcal{M}_{j+1}$, since S and Y can get from $\mathcal{M}_j$ to $\mathcal{M}_{j+1}$ only through a state C^b on the boundary of $\mathcal{M}_j$ and $\mathcal{M}_{j+1}$. The number of transitions from $\mathcal{M}_j$ to $\mathcal{M}_{j+1}$ equals $\Delta - i$. For each state C^b on the path, the ratio of

transition probabilities from C^b to the next state $C \cup \{x\}$ for Markov chains S and Y equals

$$\frac{H(X,C)}{U(P,C)+H(X,C)} \leq \frac{\widetilde{H}(X,C'')}{U(P,C'')+\widetilde{H}(X,C'')},$$

here we used that (a) $U(P,C) \geq U(P,C'')$ since $U_t(P)$ is a non-increasing process; and (b) $H(P,C) \leq \widetilde{H}(P,C'')$ since $H_t(P) \leq \widetilde{H}_{t'}(P)$ if $t \leq t'$ (see Section 2.1). □

We now prove an analog of Corollary 2.5 for $\widetilde{H}(X,Y_j)$.

Lemma 2.9. *$\widetilde{H}(X,Y_t)$ is a supermartingale.*

Proof. If $Y_j = C$, then Y_{j+1} can only be in $\{C^a, C^b\}$. Since $\widetilde{H}(X,C^a) = \widetilde{H}(X,C^b) = \widetilde{H}(X,C)$, we have $\mathbb{E}[\widetilde{H}(X,Y_{j+1}) \mid Y_j = C] = \widetilde{H}(X,Y_j)$.

If $Y_j = C^a$, then $Y_{j+1} = C' = C \cup \{c\}$ where the new center c should be in uncovered clusters with respect to C_t. We have

$$\mathbb{E}[H(P',Y_{j+1}) \mid Y_j = C^a, c \in P'] \leq 5\mathrm{OPT}_1(P'),$$

which implies

$$\mathbb{E}[\widetilde{H}(P',Y_{j+1}) \mid Y_j = C^a, c \in P'] \leq \widetilde{H}(P',Y_j).$$

Therefore, we have

$$\mathbb{E}[\widetilde{H}(X,Y_{j+1}) \mid Y_j = C^a] \leq \widetilde{H}(X,Y_j).$$

If $Y_j = C^b$, then for any possible state C' of Y_{j+1}, the new center should be in covered clusters with respect to C. By definition, we must have $\widetilde{H}(X,C') = \widetilde{H}(X,C) = \widetilde{H}(X,C^b)$. Thus, it holds that $\mathbb{E}[\widetilde{H}(X,Y_{j+1}) \mid Y_j = C^b] = \widetilde{H}(X,Y_j)$.

Combining all these cases, we get $\widetilde{H}(X,Y_j)$ is a supermartingale. □

We now use Lemma 2.8 and Lemma 2.9 to bound the expected uncovered cost of P after $k+\Delta$ rounds of k-means++.

Lemma 2.10. *For any cluster $P \in \mathcal{P}$ and $t \leq k+\Delta$, we have*

$$\mathbb{E}[U_{k+\Delta}(P) \mid C_t] \leq \frac{\widetilde{H}_t(X)}{e(\Delta - M(C_t) - 1)}.$$

Proof. Since k-means++ samples $k+\Delta$ centers and the total number of clusters in the optimal solution $\mathcal{P}$ is k, k-means++ must make Δ misses. Hence, the process X_t which follows k-means++ must either visit a state in $\mathcal{M}_{\geq\Delta}$ or stop in $\mathcal{H}$ (recall that we stop process X_t if it reaches $\mathcal{H}$).

If X_t stops in group $\mathcal{H}$, then the cluster P is covered which means that $U_{k+\Delta}(P) = 0$. Let $\partial\mathcal{M}_\Delta$ be the frontier of $\mathcal{M}_\Delta$ i.e., the states that X_t visits first when it reaches $\mathcal{M}_\Delta$ (recall that the transition graph of X_t is a tree). The expected cost $\mathbb{E}[U_{k+\Delta}(P) \mid C_t]$ is upper bounded by the expected uncovered cost of P at time when C_t reaches $\mathcal{M}_\Delta$. Thus,

$$\mathbb{E}[U_{k+\Delta}(P) \mid C_t] \leq \sum_{C \in \mathcal{M}_\Delta} \Pr(C \in \{X_j\} \mid C_t) U(P, C).$$

Observe that by Lemma 2.8, for any $C \in \mathcal{M}_\Delta$, we have

$$\Pr(C \in \{X_j\} \mid C_t) U(P, C) \leq \Pr(C \in \{Y_j\} \mid C_t) \cdot \left(\frac{\widetilde{H}(X, C)}{\widetilde{H}(X, C) + U(P, C)} \right)^{\Delta'} \cdot U(P, C).$$

Let $f(x) = x(1/(1+x))^{\Delta'}$. Then, $f(x)$ is maximized at $x = 1/(\Delta' - 1)$ and the maximum

value $f(1/(\Delta'-1)) = 1/(e(\Delta'-1))$. Therefore, for every $C \in \mathcal{M}_\Delta$, we have

$$\begin{aligned}\Pr(C \in \{X_j\} \mid C_t)U(P,C) &\le \Pr(C \in \{Y_j\} \mid C_t) f\left(\frac{U(P,C)}{\widetilde{H}(X,C)}\right)\widetilde{H}(X,C) \\ &\le \Pr(C \in \{Y_j\} \mid C_t)\frac{\widetilde{H}(X,C)}{e(\Delta'-1)}.\end{aligned}$$

Let $\tau = \min\{j : Y_j \in \mathcal{M}_\Delta\}$ be the stopping time when Y_j first visits $\mathcal{M}_\Delta$. We get

$$\sum_{C \in \mathcal{M}_\Delta} \Pr(C \in \{Y_j\} \mid C_t)\widetilde{H}(X,C) = \mathbb{E}[\widetilde{H}(X,Y_\tau) \mid C_t].$$

By Lemma 2.9, $\widetilde{H}(X,Y_j)$ is a supermartingale. Thus, by the optional stopping theorem,

$$\mathbb{E}[\widetilde{H}(X,Y_\tau) \mid C_t] \le \widetilde{H}(X,C_t).$$

Therefore, we have

$$\mathbb{E}[U_{k+\Delta}(P) \mid C_t] \le \frac{\widetilde{H}_t(X)}{e(\Delta - M(C_t) - 1)},$$

This concludes the proof. □

We now add up bounds from Lemma 2.10 with $t = 0$ for all clusters $P \in \mathcal{P}$ and obtain Lemma 2.7.

2.3.2 Small Number of Extra Centers

In this section, we give another bi-criteria approximation guarantee for k-means++.

Lemma 2.11. *Let* $\mathrm{cost}_{k+\Delta}(X)$ *be the cost of the the clustering resulting from sampling* $k+\Delta$

centers according to the k-means++ algorithm (for $\Delta \in \{1, \dots, 2k\}$). Then,

$$\mathbb{E}[\mathrm{cost}_{k+\Delta}(X)] \leq 5 \left(2 + \frac{1}{2e} + \ln \frac{2k}{\Delta}\right) \mathrm{OPT}_k(X).$$

Proof. Consider k-means++ clustering algorithm and the corresponding random process C_t. Fix a $\kappa \in \{1, \dots, k\}$. Let τ be the first iteration[1] (stopping time) when $K(C_\tau) \leq \kappa$ if $K(C_k) \leq \kappa$; and $\tau = k$, otherwise. We refer the reader to Section 2.1 for definitions of $M(C_t)$, $U_t(X) = U(X, C_t)$, and $K(C_t)$.

We separately analyze the cost of uncovered clusters after the first τ steps and the last $k' - \tau$ steps, where $k' = k + \Delta$ is the total number of centers chosen by k-means++.

The first step of our proof follows the analysis of k-means++ by Dasgupta (2013), and by Arthur and Vassilvitskii (2007). Define a potential function Ψ (see Dasgupta 2013):

$$\Psi_t := \frac{M(C_t)U(X, C_t)}{K(C_t)}.$$

If $K(C_t) = 0$, then $M(C_t)$ and $U(X, C_t)$ must be 0 and we let $\Psi_t = 0$

We use the following result by Dasgupta (2013) to estimate $\mathbb{E}[\Psi_\tau(X)]$ in Lemma 2.13.

Lemma 2.12 (Dasgupta (2013))**.** *For any $0 \leq t \leq k$, we have*

$$\mathbb{E}[\Psi_{t+1} - \Psi_t \mid C_t] \leq \frac{H(X, C_t)}{K(C_t)}.$$

Lemma 2.13. *Then, the following bound holds:*

$$\mathbb{E}[\Psi_\tau(X)] \leq 5 \left(1 + \ln \frac{k}{\kappa + 1}\right) \mathrm{OPT}_k(X).$$

[1] Recall, that $K(C_t)$ is a non-increasing stochastic process with $K(C_0) = k$.

Proof. Note that $\Psi_1 = 0$ as $M(C_1) = 0$. Thus,

$$\mathbb{E}[\Psi_\tau] \leq \sum_{t=1}^{\tau-1} \mathbb{E}\big[\Psi_{t+1} - \Psi_t\big] \leq \mathbb{E}\Big[\sum_{t=1}^{\tau-1} \frac{H(X, C_t)}{K(C_t)}\Big].$$

Using the inequality $H(X, C_t) \leq \widetilde{H}_k(X)$ (see Section 2.1), we get:

$$\mathbb{E}[\Psi_\tau] \leq \mathbb{E}\Big[\sum_{t=1}^{\tau-1} \frac{\widetilde{H}_k(X)}{K(C_t)}\Big] \leq \mathbb{E}\Big[\widetilde{H}_k(X) \cdot \sum_{t=1}^{\tau-1} \frac{1}{K(C_t)}\Big].$$

Observe that $K(C_1), \ldots, K(C_{\tau-1})$ is a non-increasing sequence in which two consecutive terms are either equal or $K(C_{i+1}) = K(C_i) - 1$. Moreover, $K(C_1) = k$ and $K(C_{\tau-1}) > \kappa$. Therefore, by Lemma 2.14 (see below), for every realization $C_0, C_1, \ldots, C_\tau$, we have:

$$\sum_{t=1}^{\tau-1} \frac{1}{K(C_t)} \leq 1 + \log {}^{k}\!/_{(\kappa+1)}.$$

Thus,

$$\mathbb{E}[\Psi_\tau] \leq (1 + \log {}^{k}\!/_{(\kappa+1)})\mathbb{E}[\widetilde{H}_k(X)] \leq 5(1 + \log {}^{k}\!/_{(\kappa+1)})\ \mathrm{OPT}_k(X).$$

This concludes the proof. □

Let $\kappa = \lfloor(\Delta - 1)/2\rfloor$. By Lemma 2.13, we have

$$\mathbb{E}\Big[\frac{M(C_\tau)U_\tau(X)}{K(C_\tau)}\Big] \leq 5\left(1 + \ln\frac{2k}{\Delta}\right)\mathrm{OPT}_k(X).$$

Since $U_t(X)$ is a non-increasing stochastic process, we have $\mathbb{E}[U_{k+\Delta}(X)] \leq \mathbb{E}[U_\tau(X)]$. Thus,

$$\mathbb{E}\Big[\frac{M(C_\tau)}{K(C_\tau)} \cdot U_{k+\Delta}(X)\Big] \leq 5\left(1 + \ln\frac{2k}{\Delta}\right)\mathrm{OPT}_k(X).$$

Our goal is to bound $\mathbb{E}[U_{k'}(X)]$. Write,

$$\mathbb{E}[U_{k'}(X)] = \mathbb{E}\Big[\frac{M(C_\tau)}{K(C_\tau)} \cdot U_{k'}(X)\Big] + \mathbb{E}\Big[\frac{K(C_\tau) - M(C_\tau)}{K(C_\tau)} \cdot U_{k'}(X)\Big].$$

The first term on the right hand side is upper bounded by $5\big(1+\ln \frac{2k}{\Delta}\big)\mathrm{OPT}_k(X)$. We now estimate the second term, which we denote by $(*)$.

Note that $K(C_t) - M(C_t) = k - t$, since the number of uncovered clusters after t steps of k-means++ equals the number of misses plus the number of steps remaining. Particularly, if $\tau = k$, we have $K(C_\tau) - M(C_\tau) = K(C_k) - M(C_k) = 0$. Consequently, if $\tau = k$, then the second term $(*)$ equals 0. Thus, we only need to consider the case, when $\tau < k$. Note that in this case $K(C_\tau) = \kappa$. By Lemma 2.7 (applied to all uncovered clusters), we have

$$\mathbb{E}[U_{k'}(X) \mid C_\tau, \tau] \leq \frac{K(C_\tau)}{e(\Delta' - 1)} \widetilde{H}_\tau(X),$$

where $\Delta' = \Delta - M(C_\tau)$.

Thus,

$$\mathbb{E}\Big[\frac{K(C_\tau) - M(C_\tau)}{K(C_\tau)} \cdot U_{k'}(X) \mid C_\tau, \tau\Big] \leq \frac{K(C_\tau) - M(C_\tau)}{K(C_\tau)} \cdot \frac{K(C_\tau)}{e(\Delta' - 1)} \cdot \widetilde{H}_\tau(X) = (**).$$

Plugging in $K(C_\tau) = \kappa$ and the expression for Δ' (see above), and using that $\kappa \leq (\Delta - 1)/2$, we get

$$(**) = \frac{\kappa - M(C_\tau)}{e(\Delta - M(C_\tau) - 1)}. \cdot \widetilde{H}_\tau(X) \leq \frac{1}{2e} \widetilde{H}_\tau(X).$$

Finally, taking the expectation over all C_τ, we obtain the bound

$$\mathbb{E}\Big[\frac{K(C_\tau) - M(C_\tau)}{K(C_\tau)} \cdot U_{k'}(X)\Big] \leq \frac{5\mathrm{OPT}_1(X)}{2e}.$$

Thus, $\mathbb{E}[U_{k'}(X)] \leq 5(1 + 1/2e + \ln 2k/\Delta)\mathrm{OPT}_k(X)$. Therefore,

$$\mathbb{E}[\mathrm{cost}_{k'}(X)] = \mathbb{E}[H_{k'}(X)] + U_{k'}(X) \leq 5\big(2 + \frac{1}{2e} + \ln\frac{2k}{\Delta}\big)\ \mathrm{OPT}_k(X).$$

□

We now prove Lemma 2.14.

Lemma 2.14. *For any $t \leq k$ integers $a_1 \geq a_2 \geq \cdots \geq a_t$ such that $a_1 = k$, $a_t > \kappa$ and $a_i - a_{i+1} \in \{0, 1\}$ for all $1 \leq i < t$, the following inequality holds*

$$\sum_{i=1}^{t} \frac{1}{a_i} \leq 1 + \ln\left(\frac{k}{\kappa+1}\right).$$

Proof. It is easy to see that the sum is maximized when $t = k$, and the sequence $a_1, \ldots, a_k$ is as follows:

$$\underbrace{\frac{1}{k}, \frac{1}{k-1}, \ldots, \frac{1}{\kappa+2}}_{(k-(\kappa+1))\text{ terms}}, \underbrace{\frac{1}{\kappa+1}, \ldots, \frac{1}{\kappa+1}}_{(\kappa+1)\text{ terms}}.$$

The sum of the first $(k - (\kappa + 1))$ terms is upper bounded by

$$\int_{1/(\kappa+1)}^{1/k} \frac{1}{x}\, dx = \ln\frac{k}{\kappa+1}.$$

The sum of the last $(\kappa + 1)$ terms is 1. □

2.4 Analysis of k-means$\|$

In this section, we give the analysis for the k-means$\|$ algorithm. Specifically, we show upper bounds on the expected cost of the solution after T rounds of k-means$\|$.

Theorem 2.15. *The expected cost of the clustering returned by k-means$\|$ algorithm after T rounds are upper bounded as follows:*

$$\textit{for } \ell < k, \qquad \mathbb{E}[\mathrm{cost}_{T+1}(X)] \leq \left(e^{-\frac{\ell}{k}}\right)^T \mathbb{E}[\mathrm{cost}_1(X)] + \frac{5\mathrm{OPT}_k(X)}{1 - e^{-\frac{\ell}{k}}};$$
$$\textit{for } \ell \geq k, \qquad \mathbb{E}[\mathrm{cost}_{T+1}(X)] \leq \left(\frac{k}{e\ell}\right)^T \mathbb{E}[\mathrm{cost}_1(X)] + \frac{5\mathrm{OPT}_k(X)}{1 - k/e\ell}.$$

Remark: For the second bound ($\ell \geq k$), the additive term $5\mathrm{OPT}_k(X)/(1-k/(e\ell)) \leq 8\mathrm{OPT}_k(X)$.

The probability that a point is sampled by k-means$\|$ is strictly greater than the probability that it is sampled by k-means$\|_{\mathrm{Pois}}$ since $1 - e^{-\lambda} < \lambda$ for all $\lambda > 0$. Thus, for every round, we can couple k-means$\|_{\mathrm{Pois}}$ and k-means$\|$ so that each point sampled by k-means$\|_{\mathrm{Pois}}$ is also sampled by k-means$\|$. Thus, the expected cost returned by k-means$\|$ is at most the expected cost returned by k-means$\|_{\mathrm{Pois}}$. In the following analysis, we show an upper bound for the expected cost of the solution returned by k-means$\|_{\mathrm{Pois}}$.

As a thought experiment, consider a modified k-means$\|_{\mathrm{Pois}}$ algorithm. This algorithm is given the set X, parameter k, and additionally the optimal solution $\mathcal{P} = \{P_1, \ldots, P_k\}$. Although this modified algorithm is useless in practice as we do not know the optimal solution in advance, it will be helpful for our analysis.

In every round t, the modified algorithm first draws independent Poisson random variables $Z_t(P_i) \sim \mathrm{Pois}(\lambda_t(P_i))$ for every cluster $i \in \{1, \ldots, k\}$ with rate $\lambda_t(P_i) = \sum_{x \in P_i} \lambda_t(x)$. Then, for each $i \in \{1, \ldots, k\}$, it samples $Z_t(P_i)$ points $x \in P_i$ with repetitions from P_i, picking every point

x with probability $\lambda_t(x)/\lambda_t(P_i)$ and adds them to the set of centers C_t. We assume that points in every set C_t are ordered in the same way as they were chosen by this algorithm.

We claim that the distribution of the output sets C_T of this algorithm is exactly the same as in the original k-means$\|_{\text{Pois}}$ algorithm. Therefore, we can analyze the modified algorithm instead of k-means$\|_{\text{Pois}}$, using the framework described in Sections 2.1.

Lemma 2.16. *The sets C_t in the original and modified k-means$\|_{\text{Pois}}$ algorithms are identically distributed.*

Proof. Consider $|P_i|$ independent Poisson point processes $N_x(a)$ with rates $\lambda_t(x)$, where $x \in P_i$ (here, we use variable a for time). Suppose we add a center x at step t of the algorithm if $N_x(t) \geq 1$. On the one hand, the probability that we choose x is equal to $1 - e^{-\lambda_t(x)}$ which is exactly the probability that k-means$\|_{\text{Pois}}$ picks x as a center at step t. On the other hand, the sum $N_{P_i} = \sum_{x \in P_i} N_x$ is a Poisson point process with rate $\lambda_t(P_i)$. Thus, the total number of jumps in the interval $[0, 1]$ of processes N_x with $x \in P_i$ is distributed as $Z_t(P_i)$. Moreover, the probability that N_x jumps at time a conditioned on the event that N_{P_i} jumps at time a is $\lambda_t(x)/\lambda_t(P_i)$. Thus, for every jump of N_{P_i}, we choose one random center x with probability $\lambda_t(x)/\lambda_t(P_i)$. □

Lemma 2.17. *For k-means$\|$ algorithm with parameter ℓ, the following bounds hold:*

$$\begin{aligned} &\textit{for } \ell < k, && \mathbb{E}[\text{cost}_{t+1}(X)] \leq e^{-\frac{\ell}{k}} \cdot \mathbb{E}[\text{cost}_t(X)] + 5\text{OPT}_k(X); \\ &\textit{for } \ell \geq k, && \mathbb{E}[\text{cost}_{t+1}(X)] \leq \frac{k}{e\ell} \cdot \mathbb{E}[\text{cost}_t(X)] + 5\text{OPT}_k(X). \end{aligned}$$

Proof. Since the expected cost returned by k-means$\|$ is at most the expected cost returned by k-means$\|_{\text{Pois}}$, we analyze the expected cost of the clustering after one step of k-means$\|_{\text{Pois}}$.

If the algorithm covers cluster P_i at round t, then at the next round, its uncovered cost equals 0. The number of centers chosen in P_i is determined by the Poisson random variable $Z_{t+1}(P_i)$.

Hence, P_i is uncovered at round $t+1$ only if $Z_{t+1}(P_i) = 0$. Since $U_t(P_i)$ is non-increasing in t and $U_t(P_i) \leq \mathrm{cost}_t(P_i)$, we have

$$\mathbb{E}[U_{t+1}(P_i) \mid C_t] \leq \Pr(Z_{t+1}(P_i) = 0) U_t(P_i) \leq \exp\left(-\frac{\ell\, \mathrm{cost}_t(P_i)}{\mathrm{cost}_t(X)}\right) \mathrm{cost}_t(P_i).$$

Define two function: $f(x) = e^{-x} \cdot x$; and $g(x) = f(x)$ for $x \in [0,1]$ and $g(x) = e^{-1}$ for $x \in [1, \infty)$. Then,

$$\mathbb{E}[U_{t+1}(X) \mid C_t] \leq \frac{k\mathrm{cost}_t(X)}{\ell} \cdot \frac{1}{k} \sum_{i=1}^{k} f\left(\frac{\ell \mathrm{cost}_t(P_i)}{\mathrm{cost}_t(X)}\right).$$

Since $g(x) \leq f(x)$, and $g(x)$ is concave for $x \geq 0$, we have

$$\mathbb{E}[U_{t+1}(X) \mid C_t] \leq \frac{k\mathrm{cost}_t(X)}{\ell} \cdot \frac{1}{k} \sum_{i=1}^{k} g\left(\frac{\ell\ \mathrm{cost}_t(P_i)}{\mathrm{cost}_t(X)}\right) \leq g\left(\frac{\ell}{k}\right) \frac{k\mathrm{cost}_t(X)}{\ell}.$$

Here, we use that $\sum_i \mathrm{cost}_t(P_i) = \mathrm{cost}_t(X)$.

Therefore, for $\ell \leq k$, we have

$$\mathbb{E}[U_{t+1}(X) \mid C_t] \leq e^{-\frac{\ell}{k}} \cdot \mathrm{cost}_t(X);$$

and for $\ell \geq k$, we have

$$\mathbb{E}[U_{t+1}(X) \mid C_t] \leq \frac{k}{e\ell} \cdot \mathrm{cost}_t(X).$$

Similar to Corollary 2.5, the process $\widetilde{H}_t(P)$ for k-means$\|_{\mathrm{Pois}}$ is also a supermartingale, which implies $\mathbb{E}[H_{t+1}(X)] \leq 5\mathrm{OPT}_k(X)$. This concludes the proof. □

Proof of Theorem 2.15. Applying the bound from Lemma 2.17 for t times, we get the following

results. For $\ell \leq k$,

$$\mathbb{E}[\text{cost}_{t+1}(X)] \leq \left(e^{-\frac{\ell}{k}}\right)^t \mathbb{E}[\text{cost}_1(X)] + 5\text{OPT}_k(X)\eta_t,$$

where $\eta_t = \sum_{j=1}^{t} \left(e^{-\frac{\ell}{k}}\right)^{j-1} < \frac{1}{1-e^{-\frac{\ell}{k}}}$.

For $\ell \geq k$,

$$\mathbb{E}[\text{cost}_{t+1}(X)] \leq \left(\frac{k}{e\ell}\right)^t \mathbb{E}[\text{cost}_1(X)] + 5\text{OPT}_k(X)\eta_t,$$

where $\eta_t = \sum_{j=1}^{t} \left(\frac{k}{e\ell}\right)^{j-1} \leq \frac{1}{1-\frac{k}{e\ell}}$. □

Corollary 2.18. *Consider a data set X with more than k distinct points. Let*

$$T = \ln \mathbb{E}\left[\frac{\text{cost}_1(X)}{\text{OPT}_k(X))}\right]$$

and $\ell > k$. Then, after T rounds of k-means‖, the expected cost of clustering $\mathbb{E}[\text{cost}_T(X)]$ is at most $9\text{OPT}_k(X)$.

CHAPTER 3

EXPLAINABLE CLUSTERING

In this chapter, we investigate the problem of *explainable* k-means and k-medians clustering which was recently introduced by Dasgupta, Frost, Moshkovitz, and Rashtchian (2020). Suppose, we have a data set which we need to partition into k clusters. How can we do it? Of course, we could use one of many standard algorithms for k-means or k-medians clustering. However, the k-means and k-medians clustering form a Voronoi diagram based on k centers, which usually have complicated boundaries. In many real-world applications, we want to find an *explainable* clustering – clustering which can be easily understood by a human being.

Dasgupta, Frost, Moshkovitz, and Rashtchian (2020) proposed to use a threshold decision tree to create an explainable clustering. A threshold decision tree is a binary space partitioning tree with k leaves. Each internal node of the threshold decision tree splits the data into two groups using a threshold cut (j, θ): on the one side of the cut, we have points x with $x_j \leq \theta$ and on the other side points x with $x_j > \theta$. Thus, every node of the tree corresponds to a rectangular region of the space. A decision tree with k leaves partitions data set X into k clusters, $P_1, \ldots, P_k$. See Figure 3.1 for an example. Dasgupta, Frost, Moshkovitz, and Rashtchian (2020) suggested that we use the standard k-medians and k-means objectives to measure the cost of the threshold decision tree. For k-medians in ℓ_1, the cost of a threshold decision tree $\mathcal{T}$ equals

$$\mathrm{cost}_{\ell_1}(X, \mathcal{T}) = \sum_{i=1}^{k} \sum_{x \in P_i} \|x - \hat{c}^i\|_1,$$

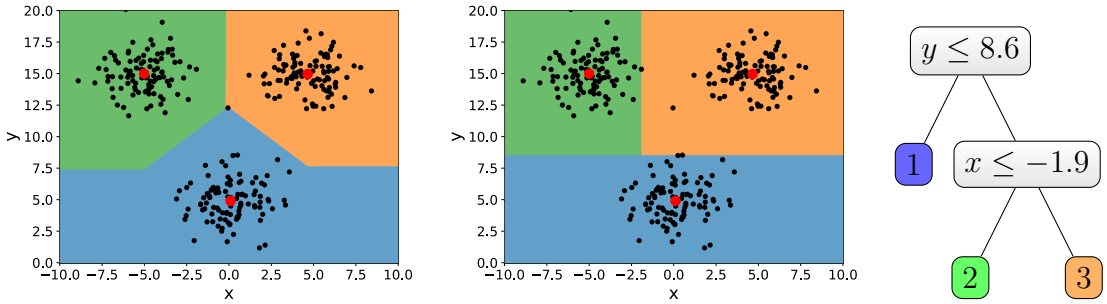


Figure 3.1: The unconstrained k-medians clustering and explainable k-medians clustering. The left diagram shows the Voronoi partition of the plane w.r.t. three centers in ℓ_1 distance. The Voronoi cell for each center consists of all points that are closer (in ℓ_1 distance) to this center than to any other center (the boundaries between cells are not straight lines because we use the ℓ_1 distance). The middle diagram shows an explainable partition. The right diagram shows the corresponding decision tree for explainable clustering.

where $P_1, \ldots, P_k$ is the partitioning of X produced by $\mathcal{T}$; and $\hat{c}^1, \ldots, \hat{c}^k$ are the medians of clusters $P_1, \ldots, P_k$. We denote the ℓ_1-norm by $\|\cdot\|_1$. Similarly, the k-means cost of a threshold tree $\mathcal{T}$ is

$$\mathrm{cost}_{\ell_2^2}(X, \mathcal{T}) = \sum_{i=1}^{k} \sum_{x \in P_i} \|x - \hat{c}^i\|_2^2,$$

where $\hat{c}^1, \ldots, \hat{c}^k$ are the means of clusters $P_1, \ldots, P_k$ given by tree $\mathcal{T}$. Note that each P_i is a rectangular region of the space. Thus, generally speaking, every x is not assigned to the closest center $\hat{c}^1, \ldots, \hat{c}^k$ like in unconstrained k-medians or k-means. Since the threshold decision tree only uses the axis-parallel cuts, the clustering given by the threshold tree is easier to understand by humans.

Dasgupta, Frost, Moshkovitz, and Rashtchian (2020) defined the price of explainability as the ratio of the k-medians cost of explainable clustering to the optimal cost of unconstrained k-medians clustering. They showed that the cost of explainability for k-means and k-medians (somewhat surprisingly) does not depend on the number of points in the data set X and only depends on

k. Specifically, they provided a greedy algorithm that given k reference centers $c^1, c^2, \cdots, c^k$ of any unconstrained k-medians as input, outputs a threshold decision tree of cost at most $O(k)$ times the cost of original unconstrained k-medians with centers $c^1, c^2, \cdots, c^k$. We call such an algorithm $O(k)$ competitive. To get an explainable k-medians clustering, we first obtain reference centers $c^1, c^2, \cdots, c^k$ using an off-the-shelf approximation algorithm for k-medians and then run an α-competitive algorithm for explainable k-medians with centers $c^1, c^2, \cdots, c^k$ given as input. This algorithm produces the desired threshold decision tree. Dasgupta, Frost, Moshkovitz, and Rashtchian (2020) also gave an $O(k^2)$ competitive algorithm for k-means and showed $\Omega(\log k)$ lower bounds on the price of explainability for both k-medians and k-means.

Our Contribution: In this chapter, we provide an algorithm RANDOMCOORDINATECUT for explainable k-medians and k-means. We show that indeed the competitive ratio of RANDOMCOORDINATECUT is at most $2 \ln k + 2$, and, therefore, this algorithm has the optimal competitive ratio which matches the lower bound of Dasgupta, Frost, Moshkovitz, and Rashtchian (2020). Our analysis is not only tight but also fairly simple. To get our result we define a game, the Set Elimination Game, which was also implicitly analyzed in previous works on this topic. We show that the cost of this game is at most $2 \ln k + 2$.

We show that this algorithm combined with a terminal embedding from ℓ_2^2 to ℓ_1 achieves a $O(k \log k)$ competitive ratio for k-means. This upper bound of the price of explainability for k-means almost matches the lower bound of $\Omega(k/\log k)$ we show in Appendix B.1. We also provide an algorithm for explainable k-medians in ℓ_2, which has an $O(\log^{3/2} k)$ competitive ratio. We complement this result with an $\Omega(\log k)$ lower bound of the price of explainability for k-medians in ℓ_2.

Organization: We first provide the algorithm for explainable k-medians in ℓ_1 and the its approximation factor in Section 3.1. Then, we provide the terminal embedding and the upper bound

Input: a data set $X \subset \mathbb{R}^d$ and set of centers $C = \{c^1, c^2, \ldots, c^k\} \subset \mathbb{R}^d$
Output: a threshold tree $\mathcal{T}$

Create tree $\mathcal{T}_0$ containing a root node r. Assign $C_r = \{c^1, c^2, \cdots, c^k\}$ to the root. Let $t = 0$. Let $M = \max_{ij} |c^i_j|$.

while $\mathcal{T}_n$ contains a leaf with at least two distinct centers **do**
Pick a random coordinate j and random $\theta \in (-M, M)$. Let $\omega_n = (j, \theta)$.
For every leaf node u in $\mathcal{T}_n$, split the set C_u into two sets:

$$\textit{Left} = \{c \in C_u : c_j \leq \theta\} \qquad \text{and} \qquad \textit{Right} = \{c \in C_u : c_j > \theta\}.$$

If both sets are not empty, then create two children of u in tree $\mathcal{T}_t$. The left child corresponds to the subregion of u with $x_j \leq \theta$, and the right child corresponds to the subregion of u with $x_j > \theta$. Assign sets *Left* and *Right* to the left and right child, respectively.

Denote the updated tree by $\mathcal{T}_{t+1}$.
Update $t = t + 1$.
end while

Figure 3.2: RANDOMCOORDINATECUT algorithm

for explainable k-means in Section 3.2. Finally, in Section 3.3, we give the algorithm for explainable k-medians in ℓ_2 and its approximation factor.

3.1 Explainable k-medians in ℓ_1

In this section, we consider the explainable clustering with k-medians in ℓ_1 objective. Dasgupta et al. (2020) introduced this problem and provided a greedy algorithm that achieves an $O(k)$ competitive ratio for k-medians in ℓ_1. The notion of explainable clustering immediately got a lot of attention in the field (Laber and Murtinho (2021); Makarychev and Shan (2021); Gamlath et al. (2021); Charikar and Hu (2022); Esfandiari et al. (2022)). Particularly, Makarychev and Shan (2021); Esfandiari, Mirrokni, and Narayanan (2022) provided almost optimal algorithms for ex-

plainable k-medians in ℓ_1, and Makarychev and Shan (2021); Esfandiari, Mirrokni, and Narayanan (2022); Gamlath, Jia, Polak, and Svensson (2021) provided almost optimal algorithms for k-means. The competitive ratios of these algorithms are $\tilde{O}(\log k)$ for k-medians and $\tilde{O}(k)$ for k-means.

The algorithms for explainable k-medians by Makarychev and Shan (2021); Esfandiari, Mirrokni, and Narayanan (2022); Gamlath, Jia, Polak, and Svensson (2021) are variants of the same simple algorithm, which we call RANDOMCOORDINATECUT. This algorithm receives a set of k reference centers $c^1, \ldots, c^k$ as input and then builds a threshold decision tree with k leaves. It works as follows. It recursively partitions d-dimensional space until every cell contains exactly one reference center c^i. The algorithm starts with a tree consisting of one node, the root. Initially, all k reference centers are assigned to that root. At every step, the algorithm picks a random threshold cut (j, θ) and splits centers in every cell using this cut. If this cut does not separate any centers in a cell u (i.e., all centers in u are located on one side of the cut), then the algorithm does not split u into two regions at this step. Finally, for every leaf u of the constructed tree, the unique center that belongs to the cell corresponding to u is assigned to u. We provide pseudo-code for this algorithm in Figure 3.2.

Makarychev and Shan (2021); Esfandiari et al. (2022) showed that the competitive ratio of RANDOMCOORDINATECUT is at most $O(\log k \log\log k)$. That is, for every data set X and set of centers $C = \{c^1, \ldots, c^k\}$,

$$\mathbb{E}[\mathrm{cost}_{\ell_1}(X, \mathcal{T})] \leq O(\log k \log\log k) \cdot \mathrm{cost}_{\ell_1}(X, C).$$

Note that the running time of this algorithm is $\tilde{O}(kd)$. Gamlath, Jia, Polak, and Svensson (2021) provided a slightly worse bound of $O(\log^2 k)$ on the competitive ratio of this algorithm. They also conjectured that this algorithm is optimal and its competitive ratio is $O(\log k)$, more specifically,

$H_{k-1} + 1$, where H_k is the k-th harmonic number. They provided some justification for their conjecture by proving this bound for a very special set of centers and data points (corresponding to the case of completely disjoint sets in our Set Elimination Game).

In this section, we show that the RANDOMCOORDINATECUT is optimal and achieves an $O(\log k)$ competitive ratio for k-medians in ℓ_1.

Theorem 3.1. *There exists a polynomial-time randomized algorithm that given a data set X and a set of centers $C = \{c^1, \ldots, c^k\}$, finds a threshold tree $\mathcal{T}$ with expected k-medians in ℓ_1 cost at most*

$$\mathbb{E}[\mathrm{cost}_{\ell_1}(X, \mathcal{T})] \leq (2 \ln k + 2) \cdot \mathrm{cost}_{\ell_1}(X, C).$$

To prove this theorem, we introduce the set elimination game in Section 3.1.1. In Section 3.1.2, we discuss the connection between explainable k-medians and set elimination games. We define a set elimination game in a set system $I \subset \{S_1, \ldots, S_k\}$ in Section 3.1.3. Then, we define the hitting and elimination time in Section 3.1.4. In Section 3.1.5, we first illustrate our proof strategy by showing Theorem 3.2 for the case when the smallest set S_1 does not overlap with $S_2, \ldots, S_k$. An important ingredient of our proof is the notion of *surprise sets*, which we discuss in Section 3.1.6. Finally, we complete the proof of Theorem 3.2 in Section 3.1.7.

3.1.1 Set Elimination Game

In this section, we define the set elimination game. Consider a finite measure space (Ω, μ) and k distinct sets $S_1, S_2, \ldots, S_k \subset \Omega$. These sets $S_1, S_2, \ldots, S_k$ may overlap with each other. The set elimination game proceeds in a series of rounds. Initially, all sets $S_1, \ldots, S_k$ enter the competition. Formally, they belong to the set of remaining sets $\mathcal{R}_0 = \{S_1, \ldots, S_k\}$. At every round n, the host picks a random $\omega_n \in \Omega$ with probability $\Pr(\omega_n = \omega) = \mu(\omega)/\mu(\Omega)$. Then, all sets S_i that contain

ω_n are eliminated from the game unless all remaining sets contain ω_n, in which case, no set gets eliminated. That is, for $n \geq 1$,

$$\mathcal{R}_n = \begin{cases} \mathcal{R}_{n-1} \setminus \{S_i \in \mathcal{R}_{n-1} : \omega_n \in S_i\}, & \text{if for some } S_i \in \mathcal{R}_{n-1}, \omega_n \notin S_i; \\ \mathcal{R}_{n-1}, & \text{otherwise.} \end{cases} \tag{3.1}$$

The last remaining set is declared the winner. We denote that winner by winner. We say that the cost of the game is the measure of the winning set, $\mu(\mathrm{winner})$.

We remark that $\mathcal{R}_n$ cannot get empty (in which case, the winner would not be defined) because of the "otherwise" clause in the definition (3.1). We shall always assume that all sets $S_1, \ldots, S_k$ are not only distinct and non-empty but also (a) for every i, $\mu(S_i) > 0$, and (b) for all i and j, $\mu(S_i \triangle S_j) > 0$ (here, $S_i \triangle S_j$ denotes the symmetric difference of sets S_i and S_j). Then, in every game, there is a unique winner with probability 1.

Our main result is the following theorem, which, as we discuss later in Section 3.1.2, implies that the competitive ratio of the explainable clustering algorithm is $2 \ln k + 2$.

Theorem 3.2. *Consider a set elimination game with the finite measure space (Ω, μ) and k distinct sets $S_1, S_2, \ldots, S_k$ (as above). The expected cost of the game is at most*

$$\mathbb{E}\big[\mu(\mathrm{winner})\big] \leq (2 \ln k + 2) \cdot \min_{i \in [k]} \mu(S_i).$$

To simplify the exposition, we will prove this theorem for discrete finite measure sets. If Ω is not a discrete measure space, we first replace it with a quotient space: We say that $\omega' \in \Omega$ and $\omega'' \in \Omega$ are equivalent ($\omega' \sim \omega''$) if they are contained in exactly the same set of sets $S_1, \ldots, S_k$. This equivalence relation partitions Ω into at most 2^k different equivalence classes. We replace Ω with the quotient space $\Omega/\sim$ whose elements are equivalence classes. In other words, we merge

all equivalent ω's. The measure of a new element $\tilde{\omega}$ equals to the measure of the corresponding equivalence class.

3.1.2 Explainable k-Medians via Set Elimination Game

In this section, we show how to use Theorem 3.2 to obtain a bound of $2\ln k + 2$ on the competitive ratio of the RANDOMCOORDINATECUT algorithm.

Theorem 3.3. *The competitive ratio of the* RANDOMCOORDINATECUT *algorithm for Explainable k-Medians is at most $2\ln k + 2$. That is, for every set of centers $C = \{c^1, \ldots, c^k\}$ and data set X, the algorithm finds a random decision tree $\mathcal{T}$ such that*

$$\mathbb{E}[\mathrm{cost}(X, \mathcal{T})] \leq (2\ln k + 2) \cdot \sum_{x \in X} \min_{c \in \{c^1, \ldots, c^k\}} \|x - c\|_1.$$

The pseudo-code for the RANDOMCOORDINATECUT algorithm is provided in Figure 3.2.

Proof. Consider an arbitrary data set $X \subset \mathbb{R}^d$ and set of k centers $C \subset \mathbb{R}^d$. We assume that all points in X and all centers in C are in the cube $[-M, M]^d$. The threshold decision tree obtained by the RANDOMCOORDINATECUT algorithm partitions the space into k cells. Each cell contains a single reference center c^i. The center c^i is not necessarily optimal for cluster P_i (cluster P_i is the intersection of the data set X and i-th cell). However, we will use it as a proxy for the optimal center. In other words, we will upper bound the cost of the threshold decision tree as follows:

$$\mathrm{cost}(X, \mathcal{T}) \equiv \min_{\hat{c}^1, \ldots, \hat{c}^k} \sum_{i=1}^{k} \sum_{x \in P_i} \|x - \hat{c}^i\|_1 \leq \sum_{i=1}^{k} \sum_{x \in P_i} \|x - c^i\|_1.$$

Let Ω be the set of all coordinate cuts: $\Omega = \{(j, \theta) : j \in [d], \theta \in [-M, M]\}$. We define a

measure μ on Ω as follows. For every subset $S \subset \Omega$, we set

$$\mu(S) = \sum_{i=1}^{d} \mu_L(\{\theta : (j, \theta) \in S\}),$$

where μ_L is the Lebesgue measure on $\mathbb{R}$. Thus, we have $\mu(\Omega) = 2dM$, which implies (Ω, μ) is a finite measure space.

Consider any data point $x \in X$. Define k sets $S_1, S_2, \ldots, S_k$ for the set elimination game. For every $i \in \{1, \ldots, k\}$, let S_i be the set of all threshold cuts that separate x and center c^i, i.e.,

$$S_i = \{(j, \theta) \in \Omega : \text{sign}(x_j - \theta) \neq \text{sign}(c^i_j - \theta)\}.$$

Note that the ℓ_1 distance from x to center c^i equals the measure of S_i: $\|x - c^i\|_1 = \mu(S_i)$. We now examine the set elimination game with sets $S_1, \ldots, S_k$, measure space (Ω, μ), and random sequence of draws $\omega_1, \omega_2, \ldots$ (each $\omega_n \in \Omega$ is the threshold cut chosen by the RANDOMCOORDINATECUT algorithm at step n). We claim that S_i belongs to $\mathcal{R}_n$ if and only if center c^i lies in the same cell as point x after step n of the algorithm. This is the case for $n = 0$, since $\mathcal{R}_0$ contains all sets $S_1, \ldots, S_k$ and the root of the threshold tree contains all centers $c^1, \ldots, c^k$. Then, whenever we pick cut ω_n, all centers separated from x by ω_n are removed from the cell of x. The only exception from this rule occurs when all centers in that cell lie on the same side of the cut ω_n. That is exactly the same rule as we have for the set elimination game (note that center c^i is separated from x by ω_n if and only if $\omega_n \in S_i$). Therefore, the same sets S_i remain in the game as center c^i in the cell of x (namely, sets S_i and centers c^i have the same indices).

The RANDOMCOORDINATECUT algorithm stops when all leaves of the decision tree contain exactly one center. At this step, the set elimination game contains one set, S_i. This set corresponds to the center c^i assigned to point x. The cost of the game $\mu(S_i)$ equals the distance from x to c^i.

By Theorem 3.2, we have

$$\mathbb{E}[\text{cost}(x, \mathcal{T})] = \mathbb{E}[\mu(\text{winner})] \leq (2 \ln k + 2) \cdot \min_i \mu(S_i) = (2 \ln k + 2) \cdot \min_i \|x - c^i\|_1.$$

We sum this bound over all data points x in X and get the desired result. □

3.1.3 Local Competitions

We now revisit the definition of the set elimination game and define competitions in subsets of $\{S_1, \ldots, S_k\}$. We remind the reader that every set elimination game is determined by an infinite sequence of i.i.d. random variables $\omega_1, \omega_2, \ldots$. For each round n and element $\omega \in \Omega$, $\Pr\big(\omega_n = \omega\big) = \mu(\omega)/\mu(\Omega)$.

Definition 3.1.1. *Consider a finite measure space (Ω, μ). Let I be a set of subsets of Ω. We say that I is a valid set system if (a) for every $S \in I$, $\mu(S) > 0$, and (b) for every $S', S'' \in I$, $\mu(S' \triangle S'') > 0$.*

The reader may assume that (Ω, μ) is a discrete finite measure space and $\mu(\omega) > 0$ for all ω in Ω. Then, the definition above says that in a valid set system I, all sets are non-empty and disjoint.

Definition 3.1.2. *Consider a finite measure space (Ω, μ). Let $\omega_1, \omega_2, \ldots$ be i.i.d. random variables as described above and I be a valid set system. We define a set elimination game in I. Initially, $\mathcal{R}_0(I) = I$. Then, for every $n \geq 1$,*

$$\mathcal{R}_n(I) = \begin{cases} \mathcal{R}_{n-1}(I) \setminus \{S \in \mathcal{R}_{n-1}(I) : \omega_n \in S\}, & \text{if for some } S' \in \mathcal{R}_{n-1}(I), \omega_n \notin S'; \\ \mathcal{R}_{n-1}(I), & \text{otherwise.} \end{cases} \tag{3.2}$$

The winner of the game in I, denoted by $\text{winner}(I)$, is the only element remaining, or, formally,

the unique element in $\cap_{n\geq 0}\mathcal{R}_n(I)$. *If* $\cap_{n\geq 0}\mathcal{R}_n(I)$ *contains more than one element, then the winner is not defined. The cost of the game is the measure of the winner,* $\mu(\text{winner}(I))$.

We remark that $\cap_{n\geq 0}\mathcal{R}_n(I)$ contains exactly one element with probability 1. Thus, the winner and cost of the game are defined with probability 1.

Consider sets $S_1, \ldots, S_k$ from Theorem 3.2. Denote $K = \{S_1, \ldots, S_k\}$. The definition of the competition among sets $S_1, \ldots, S_k$ (given in the beginning of Section 3.1.1) is exactly the same as the definition of competition in K. Our goal is to show that $\mathbb{E}[\mu(\text{winner}(K))] \leq 2(\ln k + 1) \cdot \min_{S_i \in K} \mu(S_i)$. In the proof of Theorem 3.2, we will consider competitions in different set systems $I \subseteq K$. We show the following key lemma.

Lemma 3.4. *Consider a partitioning of the set system* $K = \{S_1, \ldots, S_k\}$ *into* m *sets* $I_1, \ldots, I_m$. *Then,* $\text{winner}(K) \in \big\{\text{winner}(I_1), \ldots, \text{winner}(I_m)\big\}$.

The proof of Lemma 3.4 relies on the following observarion.

Lemma 3.5. *Let* X *and* Y *be two subsets of* K. *If* $X \subset Y$, *then for every* n, *we always have*

$$\mathcal{R}_n(Y) \cap X = \mathcal{R}_n(X) \quad \textit{or} \quad \mathcal{R}_n(Y) \cap X = \varnothing. \tag{3.3}$$

Proof. We prove that (3.3) holds by induction on n. Initially, when $n = 0$, we have $\mathcal{R}_0(X) = X$ and $\mathcal{R}_0(Y) = Y$. Therefore, $\mathcal{R}_0(Y) \cap X = X \cap Y = X = \mathcal{R}_0(X)$. Suppose (3.3) holds for n, we prove that (3.3) also holds for $n' = n + 1$. If $\mathcal{R}_n(Y) \cap X = \varnothing$, then $\mathcal{R}_n(Y) \cap X$ remains empty for all $n' \geq n$. Therefore, (3.3) holds for $n + 1$. So, let us assume that $\mathcal{R}_n(Y) \cap X = \mathcal{R}_n(X)$. Consider three cases:

- If ω_{n+1} belongs to all sets in $\mathcal{R}_n(Y)$, then it also belongs to all sets in $\mathcal{R}_n(X) = \mathcal{R}_n(Y) \cap X$. Thus, in this case, no set is eliminated in X or Y. That is, $\mathcal{R}_{n+1}(X) = \mathcal{R}_n(X)$ and $\mathcal{R}_{n+1}(Y) = \mathcal{R}_n(Y)$.

- If ω_{n+1} belongs to all sets in $\mathcal{R}_n(X)$, but not all sets in $\mathcal{R}_n(Y)$, then, at step $n+1$, we remove all sets that contain ω_{n+1} and, particularly, all sets in $\mathcal{R}_n(X)$, from $\mathcal{R}_n(Y)$. Consequently, $\mathcal{R}_{n+1}(Y) \cap X = \varnothing$.
- If not all sets in $\mathcal{R}_n(X)$ and not all sets in $\mathcal{R}_n(Y)$ contain ω_{n+1}, then we remove exactly the same sets from both $\mathcal{R}_n(X)$ and $\mathcal{R}_n(Y) \cap X$. Namely, we remove sets $S_i \in \mathcal{R}_n(Y)$ that contain ω_{n+1}.

We conclude that (3.3) holds for $n' = n + 1$. □

Proof of Lemma 3.4. Consider an arbitrary realization of the game $\omega_1, \omega_2, \ldots$. Let n be the round when all sets but the winner are eliminated from the competition i.e., $\mathcal{R}_n$ contains only one set, the winner. Since K is the union of $I_1, \ldots, I_k$, the winner must belong to some I_j. Now, by Lemma 3.5 for $X = I_j$ and $Y = K$, we have $\mathcal{R}_n(K) \cap I_j = \mathcal{R}_n(I_j)$ or $\mathcal{R}_n(K) \cap I_j = \varnothing$. We know that $\mathcal{R}_n(K) = \{\text{winner}(K)\}$ and $\text{winner}(K) \in I_j$. Thus, $\mathcal{R}_n(K) \cap I_j = \{\text{winner}(K)\} \neq \varnothing$, and

$$\mathcal{R}_n(I_j) = \mathcal{R}_n(K) \cap I_j = \{\text{winner}(K)\}.$$

We conclude that at round n, $\mathcal{R}_n(I_j)$ contains only one set – the winner in K. Consequently, it is also the winner in I_j i.e., $\text{winner}(I_j) = \text{winner}(K)$. This finishes the proof. □

3.1.4 Set Elimination with Exponential Clock

Consider a set elimination game on sets $S_1, \ldots, S_k$. It is determined by the sequence of random i.i.d. draws $\omega_1, \omega_2, \ldots$. Random variable ω_n is chosen in round n. We assign every round a random time τ_n. Let the time between two consecutive rounds be an exponential random variable with parameter $\mu(\Omega)$. Specifically, let $\Delta\tau_1, \Delta\tau_2, \ldots$ be a sequence of i.i.d. exponential random

variables with parameter $\mu(\Omega)$ and each $\tau_n = \tau_{n-1} + \Delta\tau_n = \Delta\tau_1 + \cdots + \Delta\tau_n$. Note that all $\Delta\tau_n$ are positive and $\tau_1, \tau_2, \ldots$ is an increasing sequence with probability 1. The number of draws that occur by time t (i.e., $N_t(\Omega) = |\{n : \tau_n \leq t\}|$) is a Poisson process with parameter $\mu(\Omega)$. We now can think of the set elimination game as follows: The host of the game observes a Poisson process with parameter $\mu(\Omega)$. Whenever the process jumps (at time τ_n), the host picks an element ω_n in Ω with probability $\Pr(\omega_n = \omega) = \mu(\omega)/\mu(\Omega)$ and eliminates some sets according to the rules of the game discussed above. Note that by assigning every round some time τ_n, we do not change the game, the winner, and the cost of the game (because the sequence of random draws $\omega_1, \omega_2, \ldots$ remains the same as before). This interpretation of the game allows us to introduce a hitting time $h(S)$ of every subset $S \subset \Omega$ with the following properties: (a) each $h(S)$ is an exponential random variable with rate $\mu(S)$; (b) hitting times of disjoint sets are mutually independent random variables.

Definition 3.1.3. *For every subset $X \subset \Omega$, the hitting time $h(X)$ is the time τ_n when the first ω_n is drawn from X: $h(X) = \min\{\tau_n : \omega_n \in X\}$. When the set contains one element ω, we will write $h(\omega)$ instead of $h(\{\omega\})$.*

We also define the elimination time of each set S_i.

Definition 3.1.4. *Consider any set elimination game with the measure space (Ω, μ) and k sets $S_1, S_2, \ldots, S_k$ in Ω. The elimination time $e(S_i)$ of set S_i is the time when set S_i is eliminated from the game, i.e., $e(S_i) = \min\{\tau_n : S_i \notin \mathcal{R}_n(K)\}$. If S_i is the winner, then we let $e(S_i) = \infty$ (because the winner is never eliminated).*

Note that $e(S_i) \geq h(S_i)$. Sometimes, $e(S_i)$ may be equal to $h(S_i)$, but $e(S_i)$ and $h(S_i)$ are not always the same. We now prove that hitting times for disjoint sets are independent. To this end, we *split* the Poisson process $N_t(\Omega) = |\{n : \tau_n \leq t\}|$. Let $N_t(\omega) = |\{n : \tau_n \leq t \text{ and } \omega_n = \omega\}|$.

It is easy to see that $N_t(\Omega) = \sum_{\omega \in \Omega} N_t(\omega)$ for every t. It is also true that each $N_t(\omega)$ is a Poisson process with parameter $\mu(\omega)$ and all $N_t(\omega)$ (for $\omega \in \Omega$) are mutually independent. This fact follows from the Coloring Theorem (see e.g., Kingman (1992), Coloring Theorem, page 53).

Theorem 3.6 (Coloring Theorem)**.** *Let Π_t be a Poisson process on the real line with rate λ. We color each event of the Poisson process randomly with one of M colors: The probability that a point receives the i-th color is p_i. The colors of different points are independent. Let $\Pi_t(i)$ be the number of events of color i in the interval $(0, t]$. Then, $\Pi_t(1), \ldots, \Pi_t(M)$ are independent Poisson processes. The rate of process $\Pi_t(i)$ is λp_i.*

Lemma 3.7. *For every $\omega \in \Omega$, $h(\omega)$ is an exponential random variable with parameter $\mu(\omega)$, and all random variables $h(\omega)$ (for $\omega \in \Omega$) are mutually independent.*

Proof. Observe that $h(\omega) = \min\{t : N_t(\omega) \geq 1\}$. Thus, $h(\omega)$ is an exponential random variable (the time of the first jump of a Poisson process) with rate $\mu(\omega)$. Also, since all $N_t(\omega)$ (for $\omega \in \Omega$) are mutually independent, all $h(\omega)$ are also mutually independent. □

Note that the set elimination game depends only on the hitting times for elements ω in Ω. This is the case because it matters only when every ω is drawn the first time. At that time – the hitting time of ω – all sets that contain ω are eliminated unless all remaining sets contain this ω. When the same ω is drawn again, it does not eliminate any new sets. Also, note that for any set $S \subset \Omega$, the hitting time $h(S) = \min_{\omega \in S} h(\omega)$. Thus, $h(S)$ is an exponential random variable with parameter $\mu(S) = \sum_{\omega \in S} \mu(\omega)$.

3.1.5 Approximation Factor

We now present the proof of our main result, Theorem 3.2. We assume without loss of generality that S_1 is the smallest set i.e., $\mu(S_1) \leq \mu(S_i)$ for all i. Then, the expected cost of the game is at

most:

$$\mu(S_1) + \sum_{i=2}^{k} \Pr\big(S_i = \text{winner}(K)\big)\mu(S_i). \tag{3.4}$$

We first provide some intuition for the proof by considering the case when S_1 does not intersect with sets $S_2, \ldots, S_k$, i.e. sets S_1 and S_i are disjoint for all $i = 2, 3, \ldots, k$. We split all sets into two groups S_1 and the rest of the sets $S_2, \ldots, S_k$. We know from Lemma 3.4 that the winner among all sets $S_1, \ldots, S_k$ is either S_1 or $\text{winner}\big(\{S_2, \ldots, S_k\}\big)$. Denote $I^- = \{S_2, \ldots, S_k\}$. Each set S_i is eliminated at time $e(S_i)$. The set S_1 is eliminated at its hitting time $h(S_1)$ unless it is the only remaining set at time $h(S_1)$ (because we are considering the case when S_1 does not overlap with other sets). Thus,

$$\text{winner}(K) = \begin{cases} S_1, & \text{if } h(S_1) > e(\text{winner}(I^-)); \\ \text{winner}(I^-), & \text{if } e(\text{winner}(I^-)) > h(S_1). \end{cases} \tag{3.5}$$

When the winner among $S_1, \ldots, S_k$ is not S_1, we consider two cases of the winner S_i: (1) S_i is a surprise set; (2) S_i is a non-surprise set.

Definition 3.1.5. *We say that S_i is a surprise set if $e(S_i) \geq h(S_1) \geq L/\mu(S_i)$, where $L = \ln k$.*

Let us examine bound (3.4). Let $Surprise$ be the set of all surprise sets. Note that $Surprise$ is a random set. Then,

$$\begin{aligned} \sum_{i=2}^{k} \Pr\big(S_i = \text{winner}(K)\big)\mu(S_i) \leq \sum_{i=2}^{k} \Pr\big(S_i = \text{winner}(K),\ S_i \notin Surprise\big) \cdot \mu(S_i) \\ + \sum_{i=2}^{k} \Pr\big(S_i \in Surprise\big) \cdot \mu(S_i). \end{aligned} \tag{3.6}$$

We show in the next section (Lemma 3.9) that the second sum is upper bounded by $\mu(S_1)$. We

now bound the first sum. For every winner S_i which is not a surprise set, we have $e(S_i) \geq h(S_1)$ (because S_i is the winner) and $h(S_1) \leq L/\mu(S_i)$ (because S_i is not a surprise set). We also have $S_i = \text{winner}(I^-)$, thus

$$\Pr\left(S_i = \text{winner}(K),\ S_i \notin \mathit{Surprise}\right) \leq \Pr\left(h(S_1) \leq L/\mu(S_i) \text{ and } S_i = \text{winner}(I^-)\right).$$

By Lemma 3.7, all hitting times $h(S_i) = \min_{\omega \in S_i} h(\omega)$ for $i \geq 2$ are independent from $h(S_1)$. Thus, $\text{winner}(I^-)$ is also independent of $h(S_1)$ ($\text{winner}(I^-)$ depends only on the hitting times for sets $S_i \in I^-$). Therefore,

$$\begin{aligned}\Pr\left(S_i = \text{winner}(K),\ S_i \notin \mathit{Surprise}\right) &\leq \Pr\left(h(S_1) \leq L/\mu(S_i)\right) \cdot \Pr\left(S_i = \text{winner}(I^-)\right) \\ &= \left(1 - e^{-L\mu(S_1)/\mu(S_i)}\right) \cdot \Pr\left(S_i = \text{winner}(I^-)\right) \leq \Pr\left(S_i = \text{winner}(I^-)\right) \cdot L \cdot \mu(S_1)/\mu(S_i).\end{aligned}$$

We combine all bounds on terms of (3.6) and get the following bound on the expected cost of the game:

$$\mu(S_1) + \sum_{i=2}^{k} \Pr\left(S_i = \text{winner}(I^-)\right) \cdot L \cdot \mu(S_1) + \mu(S_1) = (L+2) \cdot \mu(S_1) = (\ln k + 2) \cdot \mu(S_1).$$

This concludes the proof of the theorem for the case when S_1 does not overlap with $S_2, \ldots, S_k$. We now analyze surprise sets.

3.1.6 Surprise Sets

In this section, we prove a bound on the probability that a set S_i is a surprise set. We no longer assume that S_1 does not intersect with other sets S_i. We first show a lemma about exponential random variables.

Lemma 3.8. *Let X and Y be two independent exponential random variables with positive parameters λ_X and λ_Y, respectively. Then, for every $T \geq 0$, we have*

$$\Pr\big(Y \geq X \geq T\big) = \frac{\lambda_X}{\lambda_X + \lambda_Y} \cdot e^{-(\lambda_X+\lambda_Y)T}. \tag{3.7}$$

Proof. The desired probability can be easily found by computing $\int_T^\infty (F_X(t) - F_X(T)) f_Y(t) dt$, where $F_X(t) = 1 - e^{-\lambda_X t}$ is the cumulative distribution function of X, and $f_Y(t) = \lambda_Y \cdot e^{-\lambda_Y t}$ is the probability density function of Y. Here, we give an alternative proof. Write,

$$\begin{aligned}\Pr\big(Y \geq X \geq T\big) &= \Pr\big(Y \geq X \;\&\; \min(X,Y) \geq T\big) \\ &= \Pr\big(X \leq Y \mid \min(X,Y) \geq T\big) \cdot \Pr\big(\min(X,Y) \geq T\big).\end{aligned}$$

We have $\Pr\big(\min(X,Y) \geq T\big) = e^{-(\lambda_X+\lambda_Y)T}$, because the minimum of two independent exponential random variables with parameters λ_X and λ_Y is an exponential random variable with parameter $\lambda_X + \lambda_Y$. Then, $\Pr\big(X \leq Y \mid \min(X,Y) \geq T\big) = \Pr\big(X \leq Y\big)$ because the exponential distribution is memoryless; and $\Pr\big(X \leq Y\big) = \lambda_X/(\lambda_X + \lambda_Y)$. □

Lemma 3.9. *For every set S_i, we have* $\Pr(S_i \text{ is surprise set}) \leq \frac{1}{k} \cdot \frac{\mu(S_1)}{\mu(S_i)}$.

Proof. First, we show that $\min(e(S_i), h(S_1)) \leq h(S_i \setminus S_1)$.

Claim 3.10. *We always have* $\min(e(S_i), h(S_1)) \leq h(S_i \setminus S_1)$.

Proof. Consider an arbitrary realization of the game and the time $t = h(S_i \setminus S_1)$ when $S_i \setminus S_1$ is hit. If by this time, S_1 has already been hit then $h(S_1) < t$. Similarly, if by this time, S_i has already been eliminated then $e(S_i) < t$. Otherwise, both S_1 and S_i are still remaining in the game at time t. Therefore, when we pick $\omega \in S_i \setminus S_1$ at time t, set S_i gets eliminated (since $\omega \in S_i$; $\omega \notin S_1$; both S_1 and S_i are remaining in the game). Thus, in this case, $e(S_i) = t$. This concludes the proof. □

If S_i is a surprise set, then $\min(e(S_i), h(S_1)) = h(S_1) \geq L/\mu(S_i)$. By Claim 3.10, we have

$$h(S_i \setminus S_1) \geq \min\big(e(S_i), h(S_1)\big) = h(S_1) \geq L/\mu(S_i).$$

Thus, $\Pr(S_i \text{ is surprise set}) \leq \Pr\Big(h(S_i \setminus S_1) \geq h(S_1) \geq L/\mu(S_i)\Big)$. By Lemma 3.8 applied to the independent exponential random variables $h(S_1)$, $h(S_i \setminus S_1)$, and time $T = L/\mu(S_i)$, we have

$$\Pr(S_i \text{ is surprise set}) \leq \frac{\mu(S_1)}{\mu(S_i \setminus S_1) + \mu(S_1)} \cdot e^{-\frac{L(\mu(S_i \setminus S_1) + \mu(S_1))}{\mu(S_i)}} \leq \frac{1}{k} \cdot \frac{\mu(S_1)}{\mu(S_i)}.$$

□

3.1.7 General Case

Proof of Theorem 3.2. We upper bound the expected cost of the game for arbitrary sets $S_1, \ldots, S_k$. As before, we assume that S_1 is the smallest set. We remind the reader that each hitting time $h(S_i)$ is an exponential random variable with parameter $\mu(S_i)$. In the proof, we will use the definitions of surprise sets (see Definitions 3.1.5). We also set $L = \ln k$.

We separately upper bound the cost of the winner depending on whether the winner is (a) set S_1, (b) surprise set, (c) non-surprise set. Write

$$\begin{aligned}
\mathbb{E}\big[\mu(\text{winner}(K))\big] &= \mathbb{E}\big[\mu(\text{winner}(K)) \cdot \mathbf{1}\{\text{winner}(K) = S_1\}\big] && \text{(a)}\\
&\quad + \mathbb{E}\big[\mu(\text{winner}(K)) \cdot \mathbf{1}\{\text{winner is surprise set}\}\big] && \text{(b)}\\
&\quad + \mathbb{E}\big[\mu(\text{winner}(K)) \cdot \mathbf{1}\{\text{winner is non-surprise set}\}\big]. && \text{(c)}
\end{aligned}$$

Term (a) is upper bounded by $\mu(S_1)$. We bound term (b) using Lemma 3.9: The probability that a set is a surprise set is at most $1/k \cdot \mu(S_1)/\mu(S_i)$. Thus, the expected total measure of all sets (not

only the surprise winner) is upper bounded by $\frac{1}{k}\sum_{i=2}^{k}\frac{\mu(S_1)}{\mu(S_i)}\mu(S_i) < \mu(S_1)$.

We now bound term (c). Define a new random variable: Let $\mathrm{cost}(\omega)$ be the cost of the winner (i.e., $\mu(S_i)$, where S_i is the winner) if (1) the winner is a non-surprise set, and (2) ω is the first element that was chosen in S_1. We let $\mathrm{cost}(\omega) = 0$, otherwise. If ω is the first element that was chosen in S_1, then $h(S_1) = h(\omega)$. So, the definition of $\mathrm{cost}(\omega)$ can be written as follows:

$$\mathrm{cost}(\omega) = \mu(\mathrm{winner}(K)) \cdot \mathbf{1}\{h(S_1) = h(\omega)\} \cdot \mathbf{1}\{\mathrm{winner}(K) \notin \mathit{Surprise}\}.$$

Since the hitting time $h(S_1)$ is finite with probability 1, the term (c) equals

$$(c) = \sum_{\omega \in S_1} \mathbb{E}[\mathrm{cost}(\omega)].$$

Lemma 3.11, which we prove below, gives a bound of $2L\mu(S_1)$ on the expression above. Combining upper bounds on terms (a), (b), and (c), we get

$$\mathbb{E}\big[\mu(\mathrm{winner}(K))\big] \leq (1 + 2L + 1)\mu(S_1) = (2\ln k + 2) \cdot \mu(S_1).$$

□

Lemma 3.11. *For every $\omega \in S_1$, we have $\mathbb{E}[\mathrm{cost}(\omega)] \leq 2L\mu(\omega)$.*

Proof. We have

$$\mathbb{E}[\mathrm{cost}(\omega)] = \mathbb{E}\Big[\mu(\mathrm{winner}(K)) \cdot \mathbf{1}\{h(S_1) = h(\omega)\} \cdot \mathbf{1}\{\mathrm{winner}(K) \notin \mathit{Surprise}\}\Big]. \tag{3.8}$$

If S_i is a non-surprise set, then $h(S_1) < L/\mu(S_i)$ or $e(S_i) < h(S_1)$. If S_i is the winner, then $e(S_i) \geq h(S_1)$. Thus, if S_i is a non-surprise winner, then $h(S_1) < L/\mu(S_i)$. This observations

gives us the following upper bound on (3.8):

$$\mathbb{E}\big[\text{cost}(\omega)\big] \leq \sum_{i=2}^{k} \mu(S_i) \cdot \Pr\Big(S_i = \text{winner}(K) \text{ and } h(\omega) = h(S_1) < L/\mu(S_i)\Big). \tag{3.9}$$

Define two set systems I_ω^- and I_ω^+ of sets S_i containing and not containing ω:

$$I_\omega^- = \{S_i : \omega \notin S_i \text{ and } i \geq 2\};$$
$$I_\omega^+ = \{S_i : \omega \in S_i \text{ and } i \geq 2\}.$$

Note that $K \equiv \{S_1, \ldots, S_k\} = \{S_1\} \cup I_\omega^- \cup I_\omega^+$. By Lemma 3.4,

$$\text{winner}(K) \in \big\{S_1, \text{winner}(I_\omega^-), \text{winner}(I_\omega^+)\big\}.$$

Observe that if S_i with $i \geq 2$ is the winner, then $S_i = \text{winner}(I_\omega^-)$ or $S_i = \text{winner}(I_\omega^+)$. We replace the condition $S_i = \text{winner}(K)$ with $S_i \in \{\text{winner}(I_\omega^-), \text{winner}(I_\omega^+)\}$ in (3.9) and get bound:

$$\mathbb{E}\big[\text{cost}(\omega)\big] \leq \sum_{i=2}^{k} \mu(S_i) \cdot \Pr\Big(S_i \in \{\text{winner}(I_\omega^-), \text{winner}(I_\omega^+)\} \text{ and } h(\omega) < \frac{L}{\mu(S_i)}\Big).$$

The key observation now is that sets $\text{winner}(I_\omega^-)$ and $\text{winner}(I_\omega^+)$ are independent of $h(\omega)$. This is the case, because sets remaining in the competitions $\mathcal{R}_n(I_\omega^-)$ and $\mathcal{R}_n(I_\omega^+)$ do not change when we select ω. The set $\mathcal{R}_n(I_\omega^-)$ does not change in the round n when ω is chosen because all sets S_i in $\mathcal{R}_n(I_\omega^-) \subset I_\omega^-$ do not contain ω. The set $\mathcal{R}_n(I_\omega^+)$ does not change in this round because all sets S_i in $\mathcal{R}_n(I_\omega^+) \subset I_\omega^+$ contain ω and consequently when ω is chosen, none of these sets is removed

from $\mathcal{R}_n(I_\omega^+)$ (otherwise, $\mathcal{R}_n(I_\omega^+)$ would become empty). Thus,

$$\mathbb{E}\big[\text{cost}(\omega)\big] \le \sum_{i=2}^{k} \mu(S_i) \cdot \Pr\left(S_i \in \{\text{winner}(I_\omega^-), \text{winner}(I_\omega^+)\}\right) \cdot \Pr\left(h(\omega) < \frac{L}{\mu(S_i)}\right).$$

Using that $h(\omega)$ is an exponential random variable with parameter $\mu(\omega)$, we get (for every i)

$$\mu(S_i) \cdot \Pr\left(h(\omega) \le \frac{L}{\mu(S_i)}\right) = \mu(S_i) \cdot \left(1 - e^{-L\frac{\mu(\omega)}{\mu(S_i)}}\right) \le \mu(S_i) \cdot L\frac{\mu(\omega)}{\mu(S_i)} = \mu(\omega)L.$$

Hence,

$$\mathbb{E}\big[\text{cost}(\omega)\big] \le \mu(\omega)L \cdot \sum_{i=2}^{k} \Pr\left(S_i \in \{\text{winner}(I_\omega^-), \text{winner}(I_\omega^+)\}\right).$$

The sum on the right hand side is at most 2. Thus, $\mathbb{E}[\text{cost}(\omega)] \le 2L\mu(\omega)$. □

3.2 Explainable k-means

In this section, we consider the explainable k-means. We show that the RANDOMCOORDINATECUT algorithm with terminal embedding achieves an $O(k \log k)$ competitive ratio. This competitive ratio almost matches the $\Omega(k/\log k)$ lower bound we show in Appendix.

Theorem 3.12. *Given a set of points X in $\mathbb{R}^d$ and a set of centers C in $\mathbb{R}^d$, the* RANDOMCOORDINATECUT *algorithm in Figure 3.2 with terminal embedding finds a threshold tree $\mathcal{T}$ with expected k-means cost at most*

$$\mathbb{E}[\text{cost}_{\ell_2^2}(X, \mathcal{T})] \le O(k \log k) \cdot \text{cost}_{\ell_2^2}(X, C).$$

To prove this theorem, we show how to construct a coordinate cut preserving terminal embedding of ℓ_2^2 (squared Euclidean distances) into ℓ_1 with distortion $O(k)$ for every set of terminals $K \subset \mathbb{R}^d$ of size k.

Let K be a finite subset of points in $\mathbb{R}^d$. We say that $\varphi : x \mapsto \varphi(x)$ is a terminal embedding of ℓ_2^2 into ℓ_1 with a set of terminals K and distortion α if for every terminal y in K and every point x in $\mathbb{R}^d$, we have

$$\|\varphi(x) - \varphi(y)\|_1 \leq \|x - y\|_2^2 \leq \alpha \cdot \|\varphi(x) - \varphi(y)\|_1.$$

Lemma 3.13. *For every finite set of terminals K in $\mathbb{R}^d$, there exists a coordinate cut preserving terminal embedding of ℓ_2^2 into ℓ_1 with distortion $8|K|$.*

Proof. We first prove a one dimensional analog of this theorem (which corresponds to the case when all points and centers are in one dimensional space).

Lemma 3.14. *For every finite set of real numbers K, there exists a cut preserving embedding $\psi_K : \mathbb{R} \to \mathbb{R}$ such that for every $x \in \mathbb{R}$ and $y \in K$, we have*

$$|\psi_K(x) - \psi_K(y)| \leq |x - y|^2 \leq 8|K| \cdot |\psi_K(x) - \psi_K(y)|. \tag{3.10}$$

Proof. Let k be the size of K and $y_1, \ldots, y_k$ be the elements of K sorted in increasing order. We first define ψ_K on points in K and then extend this map to the entire real line $\mathbb{R}$. We map each y_i to z_i defined as follows: $z_1 = 0$ and for $i = 2, \ldots, k$,

$$z_i = \frac{1}{2} \sum_{j=1}^{i-1} (y_{j+1} - y_j)^2.$$

Now consider an arbitrary number x in $\mathbb{R}$. Let y_i be the closest point to x in K. Let $\varepsilon_x = \operatorname{sign}(x - y_i)$. Then, $x = y_i + \varepsilon_x |x - y_i|$. Note that $\varepsilon_x = 1$ if x is on the right to y_i, and $\varepsilon_x = -1$, otherwise. Let the function ψ_K be

$$\psi_K(x) = z_i + \varepsilon_x (x - y_i)^2.$$

For $x = (y_i + y_{i+1})/2$, both y_i and y_{i+1} are the closest points to x in K. In this case, we have

$$z_i + \varepsilon_x (x - y_i)^2 = z_{i+1} + \varepsilon_x (x - y_{i+1})^2,$$

which means $\psi_K(x)$ is well-defined.

An example of the terminal embedding function $\psi_K(x)$ is shown in Figure 3.3. Then, we show that this function ψ_K is a cut preserving embedding satisfying inequality (3.10).

We first show that this function ψ_K is continuous and differentiable in $\mathbb{R}$. Consider $2k$ open intervals on the real line divided by points in K and points $(y_i + y_{i+1})/2$ for $i \in \{1, 2, \cdots, k-1\}$. In every such open interval, the function ψ_K is a quadratic function, which is continuous and differentiable. Since ψ_K is also continuous and differentiable at the endpoints of these intervals, the function ψ_K is continuous and differentiable in . For any $x \in \mathbb{R}$, we have $\psi'_K(x) = 2|x - y^*| \geq 0$ where y^* is the closest point in K to x. Thus, the function ψ_K is increasing in $\mathbb{R}$, which implies ψ_K is cut preserving.

We now prove that ψ_K satisfies two inequalities. We first show that for every $x \in \mathbb{R}$ and $y \in K$, $|\psi_K(x) - \psi_K(y)| \leq |x - y|^2$. Suppose that $x \geq y$ (The case $x \leq y$ is handled similarly.) If $x = y$, then this inequality clearly holds. Thus, to prove $|\psi_K(x) - \psi_K(y)| \leq |x - y|^2$, it is sufficient to prove the following inequality on derivatives

$$(\psi_K(x) - \psi_K(y))'_x \leq \left((x - y)^2\right)'_x.$$

Let y^* be the closest point in K to x. Then,

$$(\psi_K(x) - \psi_K(y))'_x = (\psi_K(x))'_x = (\psi_K(y^*) + \varepsilon_x (x - y^*)^2)'_x = 2|x - y^*|.$$

Since y^* is the closest point in K to x, we have $|x - y^*| \leq |x - y| = \big((x-y)^2\big)'_x/2$. This finishes the proof of the first inequality.

We now verify the second inequality. First, consider two points y_i and y_j ($y_i < y_j$). Write,

$$\psi_K(y_j) - \psi_K(y_i) = z_j - z_i = \frac{1}{2}\sum_{m=i}^{j-1}(y_{m+1} - y_m)^2.$$

By the arithmetic mean–quadratic mean inequality, we have

$$(j-i)\cdot\sum_{m=i}^{j-1}(y_{m+1} - y_m)^2 \geq \Big(\sum_{m=i}^{j-1} y_{m+1} - y_m\Big)^2 = (y_j - y_i)^2.$$

Thus,

$$\psi_K(y_j) - \psi_K(y_i) \geq \frac{(y_j - y_i)^2}{2(j-i)} \geq \frac{(y_j - y_i)^2}{2(k-1)}.$$

Now we consider the case when x is an arbitrary real number in $\mathbb{R}$ and $y \in K$. Let y^* be the closest point in K to x. Then,

$$|x - y|^2 \leq 2|x - y^*|^2 + 2|y^* - y|^2.$$

The first term on the right hand side equals $4|\psi_K(x) - \psi_K(y^*)|$; the second term is upper bounded by $4(k-1)|\psi_K(y) - \psi_K(y^*)|$. Thus,

$$|x - y|^2 \leq 4|\psi_K(x) - \psi_K(y^*)| + 4(k-1)|\psi_K(y^*) - \psi_K(y)|.$$

Note that $|\psi_K(x) - \psi_K(y^*)| \leq |\psi_K(x) - \psi_K(y)|$ since y^* is the closest point in K to x. Also, we

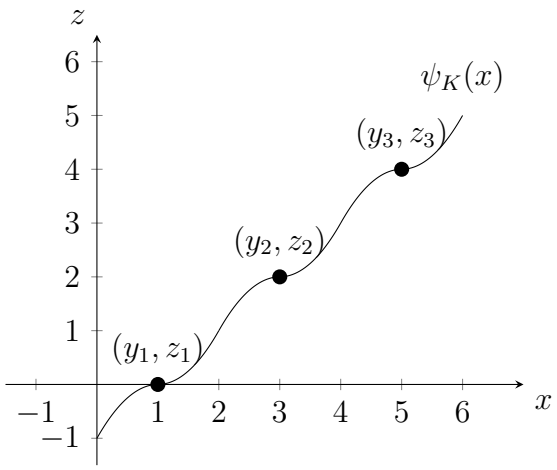


Figure 3.3: Terminal embedding function $\psi_K(x)$ for $K = \{1, 3, 5\}$.

have

$$|\psi_K(y^*) - \psi_K(y)| \leq |\psi_K(x) - \psi_K(y^*)| + |\psi_K(x) - \psi_K(y)| \leq 2|\psi_K(x) - \psi_K(y)|.$$

Hence,

$$|x - y|^2 \leq 8k|\psi_K(x) - \psi_K(y)|.$$

This completes the proof. □

Using the above lemma, we can construct a terminal embedding ψ from d-dimensional ℓ_2^2 into d-dimensional ℓ_1 as follows. For each coordinate $i \in \{1, 2, \cdots, d\}$, let K_i be the set of the i-th coordinates for all terminals in K. Define one dimensional terminal embeddings ψ_i for all coordinates i. Then, ψ maps every point $x \in \ell_2^2$ to $\psi(x) = (\psi_1(x), \cdots, \psi_d(x))$.

We show that this terminal embedding ψ is coordinate cut preserving. By the construction of φ, we have for any threshold cut (i, θ)

$$\{x \in \mathbb{R}^d : \psi(x)_i \leq \theta\} = \{x \in \mathbb{R}^d : \psi_i(x_i) \leq \theta\}.$$

Since ψ_i is a cut preserving terminal embedding by Lemma 3.14, there exists a threshold $\theta' \in \mathbb{R}$ such that

$$\{x \in \mathbb{R}^d : x_i \leq \theta'\} = \{x \in \mathbb{R}^d : \psi_i(x_i) \leq \theta\},$$

which implies ψ is coordinate cut preserving. □

For explainable k-means clustering, we first use the terminal embedding of ℓ_2^2 into ℓ_1. Then, we apply the RANDOMCOORDINATECUT algorithm to the instance after the embedding. By using this terminal embedding and Theorem 3.1, we can prove the following upper bound for explainable k-means.

Proof of Theorem 3.12. Let φ be the terminal embedding of ℓ_2^2 into ℓ_1 with terminals C. Let $\mathcal{T}'$ be the threshold tree returned by the RANDOMCOORDINATECUT algorithm on the instance after embedding. Since the terminal embedding φ is coordinate cut preserving, the threshold tree $\mathcal{T}'$ also provides a threshold tree $\mathcal{T}$ on the original k-means instance. Let $\varphi(C)$ be the set of centers after embedding. For any point $x \in X$, the expected cost of x is at most

$$\begin{aligned}
\mathbb{E}[\mathrm{cost}_{\ell_2^2}(x, \mathcal{T})] &\leq 8k \cdot \mathbb{E}[\mathrm{cost}_{\ell_1}(\varphi(x), \mathcal{T}')] \\
&\leq O(k \log k) \cdot \mathrm{cost}_{\ell_1}(\varphi(x), \varphi(C)) \\
&\leq O(k \log k) \cdot \mathrm{cost}_{\ell_2^2}(x, C),
\end{aligned}$$

where the first and third inequality is from the terminal embedding in Lemma 3.13 and the second inequality is due to Theorem 3.1. □

3.3 Explainable k-medians in ℓ_2

In this section, we present an algorithm for explainable k-medians in ℓ_2. We show that it takes a set C of k centers as input and outputs an explainable clustering with the cost at most $O(\log^{3/2} k)$ times the k-medians in ℓ_2 cost given by centers C.

Theorem 3.15. *There exists a polynomial-time randomized algorithm that given a data set X and a set of centers $C = \{c^1, \dots, c^k\}$, finds a threshold tree $\mathcal{T}$ with expected k-medians in ℓ_2 cost at most*

$$\mathbb{E}[\mathrm{cost}_{\ell_2}(X, \mathcal{T})] \leq O(\log^{3/2} k) \cdot \mathrm{cost}_{\ell_2}(X, C).$$

3.3.1 Algorithm

Our algorithm builds a binary threshold tree $\mathcal{T}$ using a top-down approach, as shown in Algorithm 3.4. It starts with a tree containing only the root node r. The root r is assigned the set of points X_r that contains all points in the data set X and all reference centers c^i. Then, the algorithm calls function BUILD_TREE(r). Function BUILD_TREE(u) partitions centers in u in several groups X_v using function PARTITION_LEAF(u) and then recursively calls itself (BUILD_TREE(v)) for every new group X_v that contains more than one reference center c^i.

Most work is done in the function PARTITION_LEAF(u). The argument of the function is a leaf node u of the tree. We denote the set of data points and centers assigned to u by X_u. Function PARTITION_LEAF(u) partitions the set of centers assigned to node u into several groups. Each group contains at most half of all centers c^i from the set X_u. When PARTITION_LEAF(u) is called, the algorithm finds the ℓ_1-median of all reference centers in node u. Denote this point by m^u. We remind the reader that the i-th coordinate of the median m^u (which we denote by m^u_i) is a median for i-th coordinates of centers in X_u. That is, for each coordinate i, both sets

Input: a data set $X \subset \mathbb{R}^d$, centers $C = \{c_1, c_2, \ldots, c_k\} \subset \mathbb{R}^d$
Output: a threshold tree $\mathcal{T}$

function MAIN(X, C)
    Create a root r of the threshold tree $\mathcal{T}$ containing $X_r = X \cup C$.
    BUILD_TREE(r).
end function

function BUILD_TREE(u)
    Call PARTITION_LEAF(u).
    Call BUILD_TREE(v) for each leaf v in the subtree of u containing more than one center.
end function

Figure 3.4: Threshold tree construction for Explainable k-medians in ℓ_2

$\{c \in X_u \cap C : c_i < m_i^u\}$ and $\{c \in X_u \cap C : c_i > m_i^u\}$ contain at most half of all centers in X_u. Then, function PARTITION_LEAF(u) iteratively partitions X_u into pieces until each piece contains at most half of all centers from X_u. We call the piece that contains the median m^u the main part (note that we find the median m^u when PARTITION_LEAF(u) is called and do not update m^u afterwards).

At every iteration t, the algorithm finds the maximum distance R_t^u from centers in the main part to the point m^u. The algorithm picks a random coordinate $i_t^u \in \{1, 2, \cdots, d\}$, random number $\theta_t^u \in [0, (R_t^u)^2]$, and random sign $\sigma_t^u \in \{\pm 1\}$ uniformly. Then, it splits the main part using the threshold cut $(i_t^u, m_i^u + \sigma_t^u \sqrt{\theta_t^u})$ if this cut separates at least two centers in the main part. Function PARTITION_LEAF(u) stops, when the main part contain at most half of all centers in X_u. Note that all pieces separated from m^u during the execution of PARTITION_LEAF(u) contain at most half of all centers in X_u because m^u is the median of all centers in X_u.

Input: a data set $X \subset \mathbb{R}^d$, centers $C = \{c_1, c_2, \ldots, c_k\} \subset \mathbb{R}^d$
Output: a threshold tree $\mathcal{T}$

function PARTITION_LEAF(u)
  Compute the ℓ_1 median m^u of all centers in X_u.
  Set the main part $u_0 = u$ and set $t = 0$.
  while node u_0 contains more than $1/2$ of centers in X_u **do**
    Update $t = t + 1$.
    Let $R_t^u = \max_{c \in X_{u_0}} \|c\|_2$.
    Sample $i_t^u \in \{1, 2, \cdots, d\}$, $\theta_t^u \in [0, (R_t^u)^2]$, and $\sigma_t^u \in \{\pm 1\}$ uniformly at random.
    if two centers in X_{u_0} are separated by $(i_t^u, m_i^u + \sigma_t^u \sqrt{\theta_t^u})$ **then**
      Assign to u_0 two children $u_\leq = \{x \in X_{u_0} : x_i \leq \vartheta\}$ and $u_> = \{x \in X_{u_0} : x_i > \vartheta\}$ where $i = i_t^u, \vartheta = m_i^u + \sigma_t^u \theta_t^u$.
      Update the main part u_0 be $u_\leq$ if $\sigma_t^u = 1$, and be $u_>$ otherwise (thus, the main part always contains m^u).
    end if
  end while
end function

Figure 3.5: Partition-Leaf Function

3.3.2 Approximation Factor

In this section, we show that our algorithm for explainable k-medians in ℓ_2 achieves an $O(\log^{3/2} k)$ competitive ratio.

Proof of Theorem 3.15. Let $\mathcal{T}_t(u)$ be the threshold tree at the beginning of iteration t in function PARTITION_LEAF(u). For every point $x \in X_u$, define its cost at step t of function PARTITION_LEAF(u) to be the distance from x to the closest center in the same leaf of $\mathcal{T}_t(u)$ as x. That is, if x belongs to a leaf node v in the threshold tree $T_t(u)$, then

$$\mathrm{cost}_{\ell_2}(x, \mathcal{T}_t(u)) = \min\{\|x - c\|_2 : c \in X_v \cap C\}.$$

If the point x is separated from its original center in C by the cut generated at time step t, then x will be eventually assigned to some other center in the main part of $\mathcal{T}_t(u)$. By the triangle inequality, the new cost of x at the end of the algorithm will be at most $\mathrm{cost}_{\ell_2}(x, C) + 2R_t^u$, where R_t^u is the maximum radius of the main part in $\mathcal{T}_t(u)$ i.e., R_t^u is the distance from the median m^u to the farthest center c^i in the main part. Define a penalty function $\phi_t^u(x)$ as follows: $\phi_t^u(x) = 2R_t^u$ if x is separated from its original center c at time t; $\phi_t^u(x) = 0$, otherwise. Let U_x be the set of all nodes u for which the algorithm calls BUILD_TREE(u) and $x \in X_u$. Note that some nodes v of the threshold tree with $x \in X_v$ do not belong to U_x. Such nodes v are created and split into two groups in the same call of PARTITION_LEAF(u). Observe that $\phi_t^u(x) \neq 0$ for at most one step t in the call of PARTITION_LEAF(u) for some node $u \in U_x$, and

$$\mathrm{cost}_{\ell_2}(x, \mathcal{T}) \leq \mathrm{cost}_{\ell_2}(x, C) + \sum_{u \in U_x} \sum_t \phi_t^u(x). \tag{3.11}$$

The sum in the right hand side is over all iterations t in all calls of function PARTITION_LEAF(u) with $u \in U_x$. Since each piece in the partition returned by function PARTITION_LEAF(u) contains at most half of all centers from X_u, the depth of the recursion tree is at most $O(\log k)$ (note that the depth of the threshold tree can be as large as $k - 1$). This means that the size of U_x is at most $O(\log k)$. In Lemma 3.17, we show that the expected total penalty in the call of PARTITION_LEAF(u) for every $u \in U_x$ is at most $O(\sqrt{\log k})$ times the original cost. Before that, we upper bound the expected penalty $\phi_t^u(x)$ for each step t in the call of PARTITION_LEAF(u) for every node $u \in U_x$.

Lemma 3.16. *The expected penalty $\phi_t^u(x)$ is upper bounded as follows:*

$$\mathbb{E}[\phi_t^u(x)] \leq \mathbb{E}\left[2\|x - c\|_2 \cdot \frac{\|c - m^u\|_2 + \|x - m^u\|_2}{d \cdot R_t^u}\right],$$

where c is the closest center to the point x in C.

Proof. We first bound the probability that point x is separated from its original center c at iteration t. For any coordinate $i \in \{1, 2, \cdots, d\}$, let x_i and c_i be the i-th coordinates of point x and center c respectively. For any point $x \in \mathbb{R}^d$, we define the indicator function $\delta_x(i, \theta) = 0$ if $x_i \leq \theta$, and $\delta_x(i, \theta) = 1$ otherwise. To determine whether the threshold cut sampled at iteration t separates x and c, we consider the following two cases: (1) x and c are on the same side of the median m^u in coordinate i (i.e. $(x_i - m_i^u)(c_i - m_i^u) \geq 0$), and (2) x and c are on the opposite sides of the median m^u in coordinate i (i.e. $(x_i - m_i^u)(c_i - m_i^u) < 0$).

If x and c are on the same side of the median m^u in coordinate i, then the threshold cut $(i, m_i^u + \sigma_t^u\sqrt{\theta_t^u})$ separates x and c if and only if σ_t^u has the same sign as $x_i - m_i^u$ and θ_t^u is between $(x_i - m_i^u)^2$ and $(c_i - m_i^u)^2$. Thus,

$$\begin{aligned}\Pr\left(\delta_x(i, \vartheta_t^u) \neq \delta_c(i, \vartheta_t^u) \mid \mathcal{T}_t(u)\right) &= \frac{|(c_i - m_i^u)^2 - (x_i - m_i^u)^2|}{2(R_t^u)^2} \\ &\leq \frac{|c_i - x_i|(|c_i - m_i^u| + |x_i - m_i^u|)}{2(R_t^u)^2},\end{aligned}$$

where $\vartheta_t^u = m_i^u + \sigma_t^u\sqrt{\theta_t^u}$.

Now, suppose x and c are on the opposite sides of the median m^u in coordinate i, i.e. $(x_i - m_i^u)(c_i - m_i^u) < 0$. The threshold cut $(i, m_i^u + \sigma_t^u\sqrt{\theta_t^u})$ separates x and c if and only if $\sigma_t^u(x_i - m_i^u) \geq 0$, $\theta_t^u \leq (x_i - m_i^u)^2$ or $\sigma_t^u(c_i - m_i^u) \geq 0$, $\theta_t^u \leq (c_i - m_i^u)^2$. Thus, we have for every coordinate i with $(x_i - m_i^u)(c_i - m_i^u) < 0$,

$$\begin{aligned}\Pr\left(\delta_x(i, \vartheta_t^u) \neq \delta_c(i, \vartheta_t^u) \mid \mathcal{T}_t(u)\right) &= \frac{(c_i - m_i^u)^2 + (x_i - m_i^u)^2}{2(R_t^u)^2} \\ &\leq \frac{|c_i - x_i|(|c_i - m_i^u| + |x_i - m_i^u|)}{2(R_t^u)^2},\end{aligned}$$

where the last inequality follows from $|c_i - x_i| \geq \max\{|c_i - m_i^u|, |x_i - m_i^u|\}$, since c_i, x_i are on the different sides of m_i^u.

Since the coordinate i_t^u is chosen randomly and uniformly from $\{1, \cdots d\}$, the probability that x and c are separated at iteration t is

$$\begin{aligned}\Pr(\delta_x(i_t^u, \vartheta_t^u) \neq \delta_c(i_t^u, \vartheta_t^u) \mid \mathcal{T}_t(u)) &\leq \sum_{i=1}^{d} \frac{|c_i - x_i|(|c_i - m_i^u| + |x_i - m_i^u|)}{2d \cdot (R_t^u)^2} \\ &\leq \frac{\|c - x\|_2(\|x - m^u\|_2 + \|c - m^u\|_2)}{d \cdot (R_t^u)^2},\end{aligned}$$

where the last inequality follows from the Cauchy-Schwarz inequality and $(|c_i|+|x_i|)^2 \leq 2c_i^2+2x_i^2$.

Then, the expected penalty is

$$\begin{aligned}\mathbb{E}[\phi_t^u(x)] &\leq \mathbb{E}\left[\Pr\left(\delta_x(i_t^u, \vartheta_t^u) \neq \delta_c(i_t^u, \vartheta_t^u) \mid \mathcal{T}_t(u)\right) \cdot 2R_t^u\right] \\ &\leq \mathbb{E}\left[2\|c - x\|_2 \cdot \frac{\|c - m^u\|_2 + \|x - m^u\|_2}{d \cdot R_t^u}\right].\end{aligned}$$

□

To bound the expected penalty for point x, we consider two types of cuts based on three parameters: the maximum radius R_t^u and distances $\|x - m^u\|_2$, $\|c - m^u\|_2$ between x, c and the median m^u . If x is separated from its original center c at iteration t with

$$R_t^u \leq \sqrt{\log_2 k} \cdot \max\{\|x - m^u\|_2, \|c - m^u\|_2\},$$

then we call this cut a light cut. Otherwise, we call it a heavy cut.

Lemma 3.17. *In every call of* PARTITION_LEAF(u) *(see Algorithm 3.4), the expected penalty for*

a point $x \in X$ is upper bounded as follows:

$$\mathbb{E}\left[\sum_t \phi_t^u(x)\right] \leq O(\sqrt{\log k}) \cdot \mathrm{cost}_{\ell_2}(x, C).$$

Proof. If point x is not separated from its original center c in PARTITION_LEAF(u), then the total penalty is 0. If x is separated from its center c in this call, then there are two cases: (1) the point x is separated by a light cut; (2) the point x is separated by a heavy cut. We first show that the expected penalty due to a heavy cut is at most $O(\sqrt{\log k})\mathrm{cost}_{\ell_2}(x, C)$.

Denote the set of all heavy cuts at iteration t in PARTITION_LEAF(u) by H_t^u:

$$H_t^u = \{x : \max\{\|x - m^u\|_2, \|c - m^u\|_2\} < R_t^u/\sqrt{\log_2 k}\}.$$

Then, by Lemma 3.16, the expected penalty x incurs due to a heavy cut is at most

$$\mathbb{E}\left[\sum_{t:x\in H_t^u} \phi_t^u(x)\right] \leq 2\|x - c\|_2 \cdot \mathbb{E}\left[\sum_{t:x\in H_t^u} \frac{\|x - m^u\|_2 + \|c - m^u\|_2}{d \cdot R_t^u}\right].$$

Since the maximum radius R_t^u is a non-increasing function of t, we split all steps of this call of PARTITION_LEAF into phases with exponentially decreasing values of R_t^u. At phase s, the maximum radius R_t^u is in the range $(R_1^u/2^{s+1}, R_1^u/2^s]$, where R_1^u is the maximum radius at the beginning of PARTITION_LEAF(u).

Consider an arbitrary phase s and step t in that phase. Let $R = R_1^u/2^s$. For every center c' with $\|c' - m^u\|_2 \in (R/2, R]$, the probability that this center c' is separated from the main part at step t in phase s is at least

$$\Pr\left(\delta_{c'}(i_t^u, \vartheta_t^u) \neq \delta_{m^u}(i_t^u, \vartheta_t^u) \mid \mathcal{T}_t(u)\right) = \sum_{j=1}^{d} \frac{1}{d} \cdot \frac{(c'_j - m_j^u)^2}{2(R_t^u)^2} = \frac{\|c' - m^u\|_2^2}{2d \cdot (R_t^u)^2} \geq \frac{1}{4d},$$

where the last inequality is due to $\|c' - m^u\|_2 > R/2 \geq R_t^u/2$ for step t in the phase s. Since there are at most k centers, all centers with norm in $(R/2, R]$ are separated from the main part in at most $4d \ln k$ steps in expectation. Thus, the expected length of each phase is $O(d \log k)$ steps, and hence, the expected penalty x incurred during phase s is at most

$$\begin{aligned}
&2\|x-c\|_2 \cdot \mathbb{E}\Bigg[\sum_{\substack{t:x\in H_t^u \\ R_t^u\in(R/2,R]}} \frac{\|x-m^u\|_2 + \|c-m^u\|_2}{d\cdot R_t^u}\Bigg] \\
&\leq 2\|x-c\|_2 \cdot \mathbb{E}\Bigg[\sum_{\substack{t:x\in H_t^u \\ R_t^u\in(R/2,R]}} \frac{\|x-m^u\|_2 + \|c-m^u\|_2}{d\cdot R/2}\Bigg] \\
&\leq O(\log k)\cdot\|x-c\|_2 \cdot \frac{\|x-m^u\|_2 + \|c-m^u\|_2}{R}.
\end{aligned}$$

Let s' be the last phase for which

$$R_1^u/2^{s'} \geq \sqrt{\log_2 k}\cdot\max\{\|x-m^u\|_2, \|c-m^u\|_2\}. \tag{3.12}$$

Then, in every phase $s > s'$, all cuts separating x from its original center c are light. Hence, the total expected penalty due to a heavy cut is upper bounded by

$$\begin{aligned}
O(\log k)\cdot\|x-c\|_2\cdot(\|x-m^u\|_2 + \|c-m^u\|_2)\cdot\sum_{s=0}^{s'}\frac{2^s}{R_1^u} = \\
= O(\log k)\cdot\|x-c\|_2\cdot(\|x-m^u\|_2 + \|c-m^u\|_2)\cdot\frac{2^{s'+1}}{R_1^u}.
\end{aligned}$$

Using the definition (3.12) of s', we write

$$(\|x-m^u\|_2 + \|c-m^u\|_2)\cdot\frac{2^{s'+1}}{R_1^u} \leq 2\frac{\|x-m^u\|_2 + \|c-m^u\|_2}{R_1^u/2^{s'}} \leq \frac{4}{\sqrt{\log_2 k}}.$$

Thus, the expected penalty due to a heavy cut is at most $O(\sqrt{\log k})\text{cost}_{\ell_2}(x, C)$.

We now analyze the expected penalty due to a light cut. Consider an iteration t in PARTITION_LEAF(u) with $x \notin H_t^u$. By the analysis in Lemma 3.16, the probability that x and c are separated at iteration t is at most

$$\frac{\|c - x\|_2(\|x - m^u\|_2 + \|c - m^u\|_2)}{d \cdot (R_t^u)^2}.$$

The probability that x or c is separated from the main part at iteration t is at least

$$\frac{\max\{\|x - m^u\|_2^2, \|c - m^u\|_2^2\}}{d(R_t^u)^2}.$$

If x or c is separated from the main part, then the point x will not incur penalty at any step after t. Thus, the probability that x and c are separated by a light cut in the end of PARTITION_LEAF(u) is at most

$$\frac{\|c - x\|_2(\|x - m^u\|_2 + \|c - m^u\|_2)}{\max\{\|x - m^u\|_2^2, \|c - m^u\|_2^2\}} \leq \frac{2\|c - x\|_2}{\max\{\|x - m^u\|_2, \|c - m^u\|_2\}}.$$

Since the penalty of a light cut is at most $R_t^u \leq \sqrt{\log_2 k} \cdot \max\{\|x - m^u\|_2, \|c - m^u\|_2\}$, the expected penalty due to a light cut is at most $O(\sqrt{\log k}) \cdot \text{cost}_{\ell_2}(x, C)$.

This concludes the proof of Lemma 3.17. □

For every node u, the main part contains the median m^u, which is also the ℓ_1-median of all centers in X_u. Thus, each cut sampled in the call PARTITION_LEAF(u) separates at most half of all centers in X_u from the origin. The main part contains at most half of centers in X_u at the end of the call PARTITION_LEAF(u). Therefore, each leaf node generated in the end of PARTITION_LEAF(u) contains at most half of centers in X_u. Thus, the depth of the recursion tree is at most $O(\log k)$. By Lemma 3.17 and Equation (3.11), we get the conclusion. □

CHAPTER 4

BI-CRITERIA APPROXIMATION FOR EXPLAINABLE k-MEANS

Explainable clustering proposed by Dasgupta et al. (2020) uses a threshold decision tree with k leaves to describe clusters. We assign a cluster to each leaf of the threshold decision tree, which corresponds to a rectangular region in the space $\mathbb{R}^d$. Recall that in the unconstrained k-means and k-medians clustering, we need to pick k centers in $\mathbb{R}^d$. The clustering forms a Voronoi partition of the space. From an information theoretical perspective, the unconstrained k-means clustering uses $(k-1)$ hyperplanes in $\mathbb{R}^d$ to describe the boundary of a cluster, which in the worst case requires $(k-1)d$ numbers. While a threshold decision tree with k leaves only uses at most $2(k-1)$ numbers to determine the boundary of each cluster. This also explains why the clustering given by a threshold decision tree is easy to understand by humans.

In Chapter 3, we provide the RANDOMCOORDINATECUT algorithm for explainable k-medians in ℓ_1 and k-means. We show that this algorithm achieves the optimal $O(\log k)$ competitive ratio for k-medians in ℓ_1 and near-optimal $O(k \log k)$ competitive ratio for k-means. However, in many real-world applications, like topic modeling or feature learning, we may use hundreds or thousands of clusters. In this case, we want to get better than $\tilde{O}(k)$ approximation on clustering cost for explainable k-means.

If we allow to expand the threshold decision tree to more than k leaves and reassign these leaves into k clusters, then we can improve the clustering cost. Frost, Moshkovitz, and Rashtchian (2020) proposed to build a threshold decision tree with more than k leaves to partition a dataset into k clusters. Each leaf in this threshold decision tree is assigned to one of the k clusters. They provided a greedy algorithm to expand a threshold decision tree and observed good experimental

results for this algorithm. However, in Section B.1.4, we provide a family of k-means instances for which the greedy bi-criteria algorithm in Frost et al. (2020) finds a threshold tree $\mathcal{T}$ with $5k/4$ leaves of cost $\mathrm{cost}_{\ell_2^2}(X, \mathcal{T}) \geq \tilde{\Omega}(k^2)\,\mathrm{OPT}_k(X)$ for $k \to \infty$.

Our Contribution: We provide a new bi-criteria algorithm for explainable k-means. Specifically, for any parameter $\delta \in (0, 1)$, our algorithm constructs a threshold decision tree with $(1+\delta)k$ leaves that achieves an $O(1/\delta \cdot \log^2 k \log\log k)$ approximation. We also provided an $\Omega(1/\delta \cdot \log^2 k)$ lower bound on the price of explainability for any threshold tree with at most $(1+\delta)k$ leaves, which means our algorithm is near-optimal. Our results characterized the trade-off between explainability and accuracy for the explainable k-means problem.

We now formally state our results. We provide a randomized algorithm for finding bi-criteria explainable k-means. Similarly to the algorithm by Frost et al. (2020), our algorithm takes k centers $\{c^1, c^2, \ldots, c^k\}$ and a parameter $\delta > 0$ and returns a threshold decision tree with $(1+\delta)k$ leaves. Each leaf of the tree is labeled with one of the centers $c^1, c^2, \ldots, c^k$. Let us denote the center returned by the decision tree $\mathcal{T}$ for point x by $\mathcal{T}(x)$. Then, the cost of explainable clustering defined by $\mathcal{T}$ equals

$$\mathrm{cost}(X, \mathcal{T}) \equiv \sum_{x \in X} \|x - \mathcal{T}(x)\|_2^2. \tag{4.1}$$

Theorem 4.1. *There exists a polynomial-time randomized algorithm that given a data set X, a set of k centers $C = \{c^1, c^2, \ldots, c^k\}$, and parameter $\delta \in (0, 1)$, creates a threshold decision tree $\mathcal{T}$ whose leaves are labeled with centers from C. The expected number of leaves in $\mathcal{T}$ is $(1+\delta)k$, and the expected cost of explainable clustering defined by $\mathcal{T}$ is*

$$\mathbb{E}[\mathrm{cost}(X, \mathcal{T})] \leq O(1/\delta \cdot \log^2 k \log\log k) \cdot \mathrm{cost}(X, C).$$

We complement our algorithmic results with an almost matching lower bound of $\Omega(1/\delta \cdot \log^2 k)$

for all threshold trees with at most $(1+\delta)k$ leaves.

Theorem 4.2. *For every $k > 500$ and $\ln^3 k/\sqrt{k} < \delta < 1/100$, there exists an instance X with k clusters such that the k-means cost for every threshold tree $\mathcal{T}$ with $(1+\delta)k$ leaves is at least*

$$\mathrm{cost}(X, \mathcal{T}) \geq \Omega\left(\frac{\log^2 k}{\delta}\right) \mathrm{OPT}_k(X).$$

4.1 Algorithm

In this section, we present an algorithm for explainable k-means clustering. The input of the algorithm is a set of centers $C = \{c^1, \ldots, c^k\}$ and a parameter $\delta \in (0, 1)$. The output is a threshold decision tree $\mathcal{T}$ in which every leaf node is labeled with one of the centers c^i. In Sections 4.3 and 4.4, we will show that the expected number of leaves in the decision tree is $(1+\delta)k$ and the approximation factor of the obtained clustering is $O(1/\delta \cdot \log^2 k \cdot \log\log k)$.

Algorithm. Our algorithm builds a binary threshold tree using a top-down approach. The algorithm assigns every node u in the tree a subset of centers $c^1, \ldots, c^k$. We denote this subset C_u. First, the algorithm creates a tree $\mathcal{T}_1$ with a root vertex r and assigns all centers $c^1, c^2, \ldots, c^k$ to it. Then, the algorithm recursively splits leaf nodes in the threshold tree until each leaf is assigned exactly one center. At each step t, the algorithm chooses a coordinate $i_t \in \{1, 2, \ldots, d\}$, a positive threshold $\theta_t \in (0, 1)$, and number σ_t in $\{\pm 1\}$ uniformly at random. For each leaf u with more than one center, it calls function *Divide-and-Share* to split node u into two parts.

Function *Divide-and-Share* first finds a median[1] of all centers assigned to u, which we denote by m^u. Let R_u be the maximum distance from centers in node u to the median m^u. The algorithm

[1]Median m^u satisfies the following property: For ever coordinate i, each of the sets $\{c \in C_u : c_i < m^u_i\}$ and $\{c \in C_u : c_i > m^u_i\}$ contains at most half of all points from C_u.

Input: a data set $X \subset \mathbb{R}^d$, a set of centers $C = \{c^1, c^2, \ldots, c^k\} \subset \mathbb{R}^d$, and a parameter $\delta \in (0, 1)$
Output: a threshold tree $\mathcal{T}$

Create a tree $\mathcal{T}_1$ containing a root r. Let $C_r = C$.

while $\mathcal{T}_t$ contains a leaf with at least two distinct centers **do**
Sample $i_t \in \{1, 2, \ldots, d\}$, $\theta_t \in (0, 1)$, and $\sigma_t \in \{\pm 1\}$ uniformly at random.

For each leaf u in the tree $\mathcal{T}_t$ containing more than one center, split node u using *Divide-and-Share* with parameters u, i_t, θ_t, σ_t, and $\varepsilon = \min\{\delta/15 \ln k, 1/384\}$.

Update $t = t + 1$.
end while

Figure 4.1: Threshold Tree Construction algorithm

creates two child nodes for u using cut $\omega_t = (i_t, \xi_t)$ with $\xi_t = m_i^u + \sigma_t \sqrt{\theta_t} R_u$. Then, *Divide-and-Share* assigns two sets of centers, $Left$ and $Right$, defined in Figure 4.2 to the left and right children of u, respectively. Note that these sets share centers in the strip of width $2\varepsilon\sqrt{\theta_t} R_u$:

$$Left \cap Right = \{c \in C_u : (m_i^u + \sigma_t \sqrt{\theta_t} R_u) - \varepsilon \sqrt{\theta_t} R_u \leq c_i \leq (m_i^u + \sigma_t \sqrt{\theta_t} R_u) + \varepsilon \sqrt{\theta_t} R_u\}.$$

If one of the sets, $Left$ or $Right$, is empty, then *Divide-and-Share* discards both newly created children of u.

We show that the bi-criteria approximation factor of the algorithm is $O(1/\delta \log^2 k \log \log k)$ and the expected number of leaves is $(1 + \delta)k$. In the next section, we give a proof overview. Then, we prove the upper bounds on the expected number of leaves and approximation factor of the algorithm in Sections 4.3 and 4.4, respectively.

Input: a node u, a coordinate $i \in \{1, \ldots, d\}$, a positive threshold θ, a number $\sigma \in \{\pm 1\}$, and a parameter ε
Output: if successful, the function splits u into two parts

Find the median of all centers assigned to node u. Denote it by m^u.

Let $R_u = \max\{\|c - m^u\|_2 : c \in C_u\}$ be the maximum distance from m^u to one of the centers in C_u.

Let

$$Left = \{c \in C_u : c_i \leq m_i^u + \sigma\sqrt{\theta}R_u + \varepsilon\sqrt{\theta}R_u\};$$
$$Right = \{c \in C_u : c_i \geq m_i^u + \sigma\sqrt{\theta}R_u - \varepsilon\sqrt{\theta}R_u\}.$$

if both sets – $Left$ and $Right$ – are nonempty **then**

Split u into two parts using cut $(i, m^u + \sigma\sqrt{\theta}R_u)$.

Assign the set of centers $Left$ to the left child u_{left} and the set of centers $Right$ to the right child, u_{right}.
end if

Otherwise, return the unmodified tree (in this case, we say that Divide-and-Share *fails*).

Figure 4.2: Function Divide-and-Share

4.2 Proof Overview

In this section, we provide an overview of the analysis of our algorithm, give definitions, and discuss the motivation for the proofs. In Sections 4.3 and 4.4, we present detailed proofs.

4.2.1 Cost of Clustering

We first analyze approximation guarantees for our algorithm. We show that the expected approximation factor is $O(1/\delta \log^2 k \log\log k) = O(1/\varepsilon \log k \log\log k)$, particularly for constant δ (e.g., $\delta = 0.05$), the expected approximation factor is $O(\log^2 k \log\log k)$. We denote the final tree returned by the algorithm by $\mathcal{T}$. Let $\mathcal{T}(x)$ be the center assigned by the threshold tree $\mathcal{T}$ to point x.

Theorem 4.3. *For every set of centers $c^1, \ldots, c^k$ in $\mathbb{R}^d$, every $\delta \in (0,1)$, and every $x \in \mathbb{R}^d$, we have*

$$\mathbb{E}\Big[\|x - \mathcal{T}(x)\|_2^2\Big] \leq O(1/\delta \ \log^2 k \log\log k) \min_{c \in \{c^1,\ldots,c^k\}} \|x - c\|_2^2. \tag{4.2}$$

This theorem guarantees that the expected approximation factor for every point x is at most $O(1/\delta \ \log^2 k \log\log k)$. Consequently, the expected approximation factor for any data set X is also bounded by $O(1/\delta \ \log^2 k \log\log k)$.

Fix an arbitrary point x for the entire proof of Theorem 4.3. If x equals one of the centers c^i, then $\mathcal{T}(x)$ also always equals c^i. Hence, $\|x - \mathcal{T}(x)\|_2^2 = 0$ and bound (4.2) trivially holds. So, from now on, we will assume that x is not one of the centers.

Denote by $\mathcal{T}_t$ the tree built by the algorithm in the first $(t-1)$ steps. Tree $\mathcal{T}_1$ contains only one node – the root. The root corresponds to the entire space $\mathbb{R}^d$ and all centers $c^1, \ldots, c^k$ are assigned to it. Since point x is fixed, we will only consider nodes u in $\mathcal{T}$ that contain x. Let u_t be the leaf node of the tree $\mathcal{T}_t$ that contains x. That is, u_t is the leaf node that contains x at the beginning of iteration t. Nodes $u_1, u_2, \ldots$ form a path in the tree $\mathcal{T}$ from the root to the unique leaf of $\mathcal{T}$ that contains x. To simplify notation, we denote

$$C_t = C_{u_t}, \quad R_t = R_{u_t}, \quad m^t = m^{u_t}.$$

Also, let D_t be the diameter of set C_t:

$$D_t = \max\{\|c' - c''\|_2 : c', c'' \in C_t\}.$$

Finally, let $\mathcal{T}_t(x)$ be the closest center from the set C_t to point x. We call this center the tentative center for point x at step t. The tentative cost of x at step t is $\|x - \mathcal{T}_t(x)\|_2^2$.

Initially, at step 1, the tentative center for point x is the closest center $c \in \{c^1, \ldots, c^k\}$ to x. If the tentative center for x does not change, then the eventual cost of x, $\|x - \mathcal{T}(x)\|_2^2$ exactly equals the optimal cost $\|x - c\|_2^2$. However, at some step t, point x may be separated from its tentative center c (see below for a formal definition), in which case another tentative center $\mathcal{T}_{t+1}(x)$ is assigned to x. At this step, the tentative cost of x may significantly increase. Moreover, the tentative cost of x may further increase if x is separated from the new tentative center. Our goal is to give an upper bound on the expected total cost increase.

Definition 4.2.1. *We say that x is separated from its tentative center $c = \mathcal{T}_t(x)$ at step t, if $c \notin C_{t+1}$.*

Note that x is separated from its tentative center $c = \mathcal{T}_t(x)$ at step t if and only if c is no longer the tentative center for x at step $t+1$ ($\mathcal{T}_{t+1}(x) \neq \mathcal{T}_t(x)$). We now define A_k. Loosely, speaking A_k is the approximation factor of the algorithm for the given set of centers $c^1, \ldots, c^k$ and point x. For technical reasons, the formal definition is more involved.

Definition 4.2.2. *Let A_k be the smallest number such that the following inequality holds with probability 1 for every partially built tree $\mathcal{T}_t$:*

$$\mathbb{E}\Big[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_t\Big] \leq A_k \, \|x - \mathcal{T}_t(x)\|_2^2. \tag{4.3}$$

In this definition, $\mathbb{E}\Big[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_t\Big]$ is the conditional expectation of the eventual cost of x

given that at step t the partially built tree is $\mathcal{T}_t$. Thus, if at some step t, the tentative center for x is c, then the expected final cost $\mathbb{E}[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_t]$ is upper bounded by $A_k \ \|x - c\|_2^2$. Observe, that A_k is well defined and finite, because $\mathcal{T}(x)$ and $\mathcal{T}_t(x)$ take at most k different values (namely, values in $\{c^1, \ldots, c^k\}$).

We show an upper bound of $O(1/\varepsilon \log k \log\log k)$ on A_k (note: $\varepsilon = \min\{\delta/15 \ln k, 1/384\}$). To illustrate the proof, we make a number of simplifying assumptions in this section. The actual proof is considerably more involved. We give it in Section 4.4.

Informal Proof of the Upper Bound on A_k. Suppose c^* is the tentative center for x at step t^*. If at some step $t \geq t^*$, center c^* is separated from x, then we assign a new tentative center to x. We call this center a *fallback* center for x. This fallback center depends on the tree $\mathcal{T}_t$ and cut (i, ξ) that separates x and c^*. However, to illustrate the idea behind the proof, let us assume that the distance from the *fallback* center to x does not depend on the cut (i, ξ). Specifically, we suppose that the distance from x to the fallback center is M_t at step t *for every cut* (i, ξ).

We consider four possibilities:

A. Point x and c^* are never separated.

B. Point x is separated from c^* at step t and $D_t^2 \leq \|x - c^*\|_2^2$.

C. Point x is separated from c^* at step t and $\|x - c^*\|_2^2 < D_t^2 \leq A_k M_t^2/2$.

D. Point x is separated from c^* at step t and $D_t^2 > A_k M_t^2/2$.

In case (A), the cost of x in the resulting tree $\mathcal{T}$ equals $\|x - c^*\|_2^2$. In cases (B) and (C), the eventual cost of x is upper bounded by $(D_t + \|x - c^*\|_2)^2 \leq 2D_t^2 + 2\|x - c^*\|_2^2$ because no matter which center c^{**} in C_t is assigned to x in $\mathcal{T}$, the distance from c^{**} to x is at most $\|x - c^*\|_2 + \|c^* - c^{**}\|_2 \leq \|x - c^*\|_2 + D_t$ (note: D_t is the maximum distance between centers

in C_t). Furthermore, in case (B), $2D_t^2 + 2\|x - c^*\|^2 \leq 4\|x - c^*\|^2$. In case (D), after step t, the distance from x to the new tentative center is M_t. Hence, by the definition of A_k (see Definition 4.2.2), the expected cost of x in $\mathcal{T}$ is bounded by $A_k M_t^2$. To summarize, in case (A) or (B), the final cost of x is at most $4\|x - c^*\|_2^2$. In case (C) and (D), the final cost is upper bounded by $2\|x - c^*\|_2^2 + \min\left\{2D_t^2, A_k M_t^2\right\}$, where t is the step when x and c^* are separated.

Let t^{**} be the first step t of the algorithm, when $D_t \leq \|x - c^*\|_2$ or c^* is no longer the tentative center for x. Note that for some step t, C_t contains only one center and $D_t = 0$. Hence, the stopping time t^{**} is well defined. Then,

$$\mathbb{E}[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_{t^*}] \leq 4\|x - c^*\|_2^2 + \\ + \mathbb{E}\left[\sum_{t=t^*}^{t^{**}-1} \Pr\{x \;\&\; c^* \text{ are separated at step } t \mid \mathcal{T}_t\} \min\left\{2D_t^2, A_k M_t^2\right\} \mid \mathcal{T}_{t^*}\right].$$

We need to estimate the probability that x and c^* are separated at step t. Observe that if x and c^* are separated, then $x_i - m_i^t \leq \sigma_t\sqrt{\theta_t}R_t$ and $c_i^* - m_i^t \geq (\sigma_t + \varepsilon)\sqrt{\theta}R_t$ or $x_i - m_i^t \geq \sigma_t\sqrt{\theta}R_t$ and $c_i^* - m_i^t \leq (\sigma_t - \varepsilon)\sqrt{\theta}R_t$, where $i = i_t$ is the coordinate chosen by the algorithm. We consider the case when x_i and c_i^* are on the same side of m_i^t, i.e. $(x_i - m_i^t)(c_i^* - m_i^t) \geq 0$. The case when x_i and c_i^* are on the opposite sides of m_i^t is handled similarly. Since θ_t is uniformly distributed in $[0, 1]$ and coordinate i_t is chosen randomly from $\{1, \ldots, d\}$, we have

$$\Pr\{x \;\&\; c^* \text{ are separated at step } t \mid \mathcal{T}_t\} \leq \\ \leq \frac{1}{d\,R_t^2} \sum_{i=1}^{d} \max\left\{\frac{|c_i^* - m_i^t|^2}{(1+\varepsilon)^2} - |x_i - m_i|^2, |x_i - m_i|^2 - \frac{|c_i^* - m_i^t|^2}{(1-\varepsilon)^2}, 0\right\}.$$

Remark: In the formula above, we divide $|c_i^* - m_i^t|^2$ by $(1+\varepsilon)^2$ and $|c_i^* - m_i^t|^2$ by $(1-\varepsilon)^2$. These factors – $1/(1+\varepsilon)^2$ and $1/(1-\varepsilon)^2$ – are essential for the analysis. If we did not have them, we would get

$\tilde{\Theta}(k)$ instead of $O(1/\varepsilon \log k \log\log k)$ approximation!

We now use the following inequality: For all positive numbers a, b and $\varepsilon \in (0,1)$, we have

$$\max\left\{\frac{b^2}{(1+\varepsilon)^2} - a^2, b^2 - \frac{a^2}{(1-\varepsilon)^2}\right\} \le \frac{(b-a)^2}{2\varepsilon - \varepsilon^2} \le \frac{(b-a)^2}{\varepsilon}. \tag{4.4}$$

This inequality can be verified by dividing the left and right hand sides by a^2 and solving the obtained quadratic equation for $\lambda = b/a$. We have

$$\Pr\{x \text{ \& } c^* \text{ are separated at step } t \mid \mathcal{T}_t\} \le \frac{1}{d\,R_t^2}\sum_{i=1}^{d}\frac{(x_i - c_i^*)^2}{\varepsilon} = \frac{\|x-c^*\|_2^2}{\varepsilon d\,R_t^2}.$$

Note that the separation probability is proportional to the squared distance between x and its tentative center c^* (i.e., $\|x-c^*\|_2^2$) rather than the distance $\|x-c^*\|_2$ itself.

In Section 4.4, we are going to use a slightly different version of inequality (4.4) to bound the probability that x and c^* are separated using a particular cut (i,ξ) (see Claim 4.14).

We use the upper bound on the separation probability to obtain a convenient bound on the expected final cost of x:

$$\begin{aligned}\mathbb{E}[\|x-\mathcal{T}(x)\|_2^2 \mid \mathcal{T}_{t^*}] &\le 4\|x-c^*\|_2^2 + \mathbb{E}\left[\sum_{t=t^*}^{t^{**}-1}\frac{\|x-c^*\|_2^2}{\varepsilon d R_t^2}\cdot\min\left\{2D_t^2, A_k M_t^2\right\} \mid \mathcal{T}_{t^*}\right]\\ &= \|x-c^*\|_2^2\cdot\left(4 + \mathbb{E}\left[\frac{1}{\varepsilon d}\sum_{t=t^*}^{t^{**}-1}\frac{\min\left\{2D_t^2, A_k M_t^2\right\}}{R_t^2} \mid \mathcal{T}_{t^*}\right]\right).\end{aligned}$$

Thus,

$$\mathbb{E}\left[\frac{\|x-\mathcal{T}(x)\|_2^2}{\|x-c^*\|_2^2} \mid \mathcal{T}_{t^*}\right] \le 4 + \mathbb{E}\left[\frac{1}{\varepsilon d}\sum_{t=t^*}^{t^{**}-1}\frac{\min\left\{2D_t^2, A_k M_t^2\right\}}{R_t^2} \mid \mathcal{T}_{t^*}\right]. \tag{4.5}$$

Our goal is to bound the right hand side of this inequality by $O(1/\varepsilon \log k \log\log k)$.

In Lemma 4.6, we show that $R_t \approx D_t$. Specifically, $1/\sqrt{2}R_t \le D_t \le 2R_t$. This inequality would

be trivial if m^t was one of the centers c^j. However, generally speaking, this is not the case. In fact, m^t does not have to belong to the convex hull of centers in C_t. Nevertheless, $D_t \in [1/\sqrt{2}R_t, 2R_t]$ because m^t is the median of C_t (see Lemma 4.6).

It is easy to see that the diameter D_t is a non-increasing function of t (since $C_{t+1} \subset C_t$) and M_t is a non-decreasing function of t. In Lemma 4.7, we show that, in fact, D_t decreases by a factor of 2 every $L = \Theta(d \ln k)$ steps with high probability. That is, $D_{t+L} \leq D_t/2$. This happens because for every step t, each pair of centers c' and c'' with $\|c' - c''\|_2 \geq D_t/2$ assigned to u_t is separated with probability at least $\Omega(1/d)$ (see Corollary 4.9). So, in $L = \Theta(d \ln k)$ steps all pairs of centers in C_t at distance at least $D_t/2$ are separated with high probability.

We upper bound the right hand side of (4.5). Write

$$\begin{aligned}\frac{1}{\varepsilon d}\sum_{t=t^*}^{t^{**}-1}\frac{\min\left\{2D_t^2, A_kM_t^2\right\}}{R_t^2} &\leq \sum_{\substack{t\in\{t^*,\cdots,t^{**}-1\}\\ A_kM_t^2\leq 2D_t^2}}\frac{A_kM_t^2}{\varepsilon dR_t^2} + \sum_{\substack{t\in\{t^*,\cdots,t^{**}-1\}\\ 2D_t^2< A_kM_t^2}}\frac{2D_t^2}{\varepsilon dR_t^2}\\ &\leq \underbrace{\sum_{\substack{t\in\{t^*,\cdots,t^{**}-1\}\\ 2D_t^2\geq A_kM_t^2}}\frac{4A_kM_t^2}{\varepsilon dD_t^2}}_{\Sigma_I} + \underbrace{\sum_{\substack{t\in\{t^*,\cdots,t^{**}-1\}\\ 2D_t^2< A_kM_t^2}}\frac{8}{\varepsilon d}}_{\Sigma_{II}}. \end{aligned} \tag{4.6}$$

Consider the first sum, Σ_I on the right hand side of (4.6). It is upper bounded by $2L$ times the maximum term in that sum, because D_t halves every L steps and therefore $(M_t/D_t)^2$ increases by 4 times every L steps. The maximum term in Σ_I is, in turn, upper bounded by $8/(\varepsilon d)$ (because $2D_t^2 \geq A_kM_t^2$ for all terms in Σ_I).

Now consider the second sum, Σ_{II} on the right hand side of (4.6). Let t' be the first step t for which $2D_t^2 < A_kM_t^2$. Using that $D_{t+L} \leq D_t/2$, we obtain the following upper bound on the

number of steps $t < t^{**}$ in Σ_{II}:

$$t^{**} - t' \leq L + L \cdot \log_2 \frac{D_{t'}}{D_{t^{**}-1}} \leq L + L \cdot \log_2 \frac{\sqrt{A_k/2}\, M_{t'}}{D_{t^{**}-1}} \leq L + L \cdot \log_2 \frac{\sqrt{A_k/2}\, M_{t^{**}-1}}{D_{t^{**}-1}}.$$

The last inequality holds because M_t is a non-decreasing function of t. Recall, that the distance to the fallback center, M_t is upper bounded by $\|x - c^*\|_2 + D_t$ for every step $t \in \{t^*, \cdots, t^{**} - 1\}$. Also, by the definition of stopping time t^{**}, for every $t < t^{**}$, we have $D_t > \|x - c^*\|_2$. Thus,

$$\frac{M_{t^{**}-1}}{D_{t^{**}-1}} \leq \frac{\|x - c^*\|_2 + D_{t^{**}-1}}{D_{t^{**}-1}} \leq 2.$$

Therefore, $t^{**} - t' \leq L \cdot (1 + \log_2 \sqrt{2A_k})$. Consequently, the second sum, Σ_I as well as $\Sigma_I + \Sigma_{II}$ are upper bounded by $O((L \log A_k)/(\varepsilon d)) = O(1/\varepsilon \log k \log A_k)$. We obtained the following bound:

$$\mathbb{E}\left[\frac{\|x - \mathcal{T}(x)\|_2^2}{\|x - c^*\|_2^2} \mid \mathcal{T}_{t^*}\right] \leq O(1/\varepsilon \log k \log A_k).$$

Therefore, $A_k \leq O(1/\varepsilon \log k \log A_k)$. This recurrence relation gives us an upper bound of $O(1/\varepsilon \log k \log\log k)$ on A_k. This concludes the proof overview of Theorem 4.2.

4.2.2 Expected Number of Leaves

We show that the expected number of leaves in the threshold tree given by our algorithm is at most $e^{\delta/2}k$. Particularly, for $\delta \in (0, 1)$, the expected number of leaves is at most $(1 + \delta)k$. We now give an overview of the analysis. We provide a complete proof in Section 4.3.

In this section, we consider the case when the space is 1-dimensional. That is, all centers and data points lie on the real line. Consider a fixed center c. Let $N_c(\mathcal{T})$ be the number of leaves in tree $\mathcal{T}$ containing c. We show that $\mathbb{E}[N_c(\mathcal{T})]$ is at most $e^{\delta/2}$.

Suppose c is assigned to node u at step t (note that c may be assigned to several nodes). Denote the total number of centers assigned to u by $k' = |C_u|$. We prove by induction on k' that the expected number of leaves to which u is assigned in the subtree rooted at u is at most $(1+5\varepsilon)^{\log_2 k'}$. If $k' = 1$, then the claim trivially holds, since u is a leaf. Assume $k' > 1$.

Our algorithm divides u into two parts u_{left} and u_{right}. One of them contains the median m^u. We call that part the main child and denote it by u'. In turn, the main child u' is also divided into two parts, one of them – denoted by u'' – is the main child of u'. We call the sequence of nodes $u, u', u'', \ldots$ the main branch rooted at u. Note that the main child always contains at least half of all centers assigned to its parent. This is the case, because m^u is the median of all centers assigned to u. Thus, the part containing m^u contains at least half of all centers in C_u, and the other (secondary) child contains at most half of all centers in C_u.

Suppose that center c is assigned to a node v in the main branch $u, u', u'', \ldots$. When v is divided into two parts, one of the following three events may occur: (1) c is assigned only to the main child of v; (2) c is assigned to both the main and secondary children of v; (3) c is assigned only to the secondary child of v. Denote these events by $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$, respectively. We estimate the number of nodes w such that c is assigned to w, and w is a secondary child of a node in the main branch. This number equals to the number of events $\mathcal{E}_2$ that occur in the main branch before the first event $\mathcal{E}_3$ occurs plus 1. If the probabilities of events $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ were the same for all nodes in the main branch containing c, the expected number above would be equal to $1/\Pr(\mathcal{E}_3 \mid \mathcal{E}_2 \cup \mathcal{E}_3)$. Without loss of generality assume that $m^u = 0$, then for $\varepsilon \le 1/10$, we have

$$\frac{1}{\Pr(\mathcal{E}_3 \mid \mathcal{E}_2 \cup \mathcal{E}_3)} = \frac{\Pr(\mathcal{E}_2 \cup \mathcal{E}_3)}{\Pr(\mathcal{E}_3)} = \frac{c^2}{(1-\varepsilon)^2 R_t^2} \Big/ \frac{c^2}{(1+\varepsilon)^2 R_t^2} = \frac{(1+\varepsilon)^2}{(1-\varepsilon)^2} \le 1 + 5\varepsilon.$$

Every secondary child w contains at most $k'/2$ centers. So, by the inductive hypothesis, the ex-

pected number of leaves containing c in the subtree rooted at w is at most $(1+5\varepsilon)^{\lfloor \log_2 k'/2 \rfloor}$. Therefore, the expected number of leaves containing c in the subree rooted at u is at most

$$(1+5\varepsilon) \cdot (1+5\varepsilon)^{\lfloor \log_2 k'/2 \rfloor} \leq (1+5\varepsilon)^{\lfloor \log_2 k' \rfloor}.$$

This concludes the proof of the inductive claim. We now observe that

$$\mathbb{E}[N_c(\mathcal{T})] \leq (1+5\varepsilon)^{\lfloor \log_2 k \rfloor} \leq e^{\delta/2}$$

for $\varepsilon \leq \frac{\delta}{15 \ln k}$.

4.3 Expected Number of Leaves

In this section, we prove a bound the expected number of leaves in the threshold tree constructed by our algorithm. Our algorithm assigns all centers $c^1, \ldots, c^k$ to the root r of the threshold tree $\mathcal{T}$. Then, it recursively divides centers assigned to every node u between its children. However, centers in a narrow strip $Left \cap Right$ are shared by the both children of node u. Thus, the total number of leaves in the threshold tree $\mathcal{T}$ may be larger than k. Let $N(\mathcal{T})$ be the number of leaves in $\mathcal{T}$. We show an upper bound of $e^{\delta/2}k$ on the expected number of leaves $\mathbb{E}[N(\mathcal{T})]$, where the expectation is over the randomness of our algorithm.

Theorem 4.4. *For every set of centers $c^1, c^2, \ldots, c^k$ in $\mathbb{R}^d$ and every $\delta \in (0, \ln k/32)$, the expected number of leaves in the threshold tree $\mathcal{T}$ given by our algorithm is at most*

$$\mathbb{E}_{\mathcal{T}}[N(\mathcal{T})] \leq e^{\delta/2}k.$$

In particular, for $\delta \in (0, 1)$*,*

$$\mathbb{E}_{\mathcal{T}}[N(\mathcal{T})] \leq (1 + \delta)k.$$

Proof. For every center c, we bound the expected number of leaves containing c by $e^{\delta/2}$. Consider a fixed center c. For a node u in the threshold tree $\mathcal{T}$, let $N_c^u(\mathcal{T})$ denote the number of leaves in the subtree of $\mathcal{T}$ rooted at node u to which center c is assigned to.

Definition 4.3.1. *For every integer* $k' \in \{1, 2, \ldots, k\}$*, let* $B_{k'}$ *be the minimum number such that the following inequality holds for every partially built tree* $\mathcal{T}_t$ *and every leaf* u *with* $|C_u| \leq k'$ *in* $\mathcal{T}_t$ *to which center* c *is assigned,*

$$\mathbb{E}[N_c^u(\mathcal{T}) \mid \mathcal{T}_t] \leq B_{k'}.$$

That is, $B_{k'}$ is an upper bound on the expected number of leaves in the subtree rooted at u that contain c if at most k' centers are assigned to u. To prove Theorem 4.4, it is sufficient to show that B_k is at most $1 + \delta$. We derive the following recurrence relation on $B_{k'}$.

Lemma 4.5. *The upper bound on the expected number of leaves* $B_{k'}$ *satisfies the following recurrence relation:*

$$B_1 = 1, \tag{4.7}$$

$$B_{k'} \leq (1 + 5\varepsilon)B_{\lfloor k'/2 \rfloor}, \tag{4.8}$$

where $\varepsilon = \min\{\delta/15 \ln k, 1/384\}$.

Proof. It is easy to see that $B_1 = 1$, because if c is the only center assigned to node u, then u is a leaf and $N_c^u(\mathcal{T}) = 1$. We now prove (4.8). Consider a partially built tree $\mathcal{T}_t$, node u in $\mathcal{T}_t$, and center c in X_u for which inequality (4.3.1) is tight i.e., $B_{k'} = \mathbb{E}[N_c^u(\mathcal{T}) \mid \mathcal{T}_t]$.

Examine the call of function *Divide-and-Share* that splits node u. Let i_t be the coordinate randomly chosen for this call of function *Divide-and-Share*. Without loss of generality, we assume that $c_i \geq m_i^u$. If σ_t is negative, then center c is assigned only to the right child of u. In this case, the expected number of leaves containing c in the subtree rooted at u is at most $B_{k'}$.

We now consider the case when $\sigma_t = 1$. Define three disjoint events: (1) center c is assigned only to the left child of u and $\sigma_t = 1$; (2) center c is assigned to both children of u and $\sigma_t = 1$; (3) center c is assigned only to the right child of u and $\sigma_t = 1$. Denote these events by $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$, respectively.

The number of centers assigned to node u is k'. Thus, the number of centers assigned to each child of u is at most k'. Moreover, if $\sigma_t = 1$, the number of centers assigned to the *right* child u_{right} of u is at most $\lfloor k'/2 \rfloor$, because m^u is the median of all centers in C_u and for all centers c' assigned to u_{right}, $c'_i > m_i^u$. Hence, if event $\mathcal{E}_1$ occurs, then the expected number of leaves containing c in the subtree rooted at u is bounded by $B_{k'}$. If event $\mathcal{E}_2$ occurs, then the expected number of leaves containing c in the subtree rooted at u is bounded by $B_{k'} + B_{\lfloor k'/2 \rfloor}$. Finally, if event $\mathcal{E}_3$ occurs, then the expected number of leaves containing c in the subtree rooted at u is bounded by $B_{\lfloor k'/2 \rfloor}$. Thus,

$$\begin{aligned}\mathbb{E}[N_c^u(\mathcal{T}) \mid \mathcal{T}_t] &\leq 1/2 B_{k'} + B_{k'} \Pr(\mathcal{E}_1 \mid \mathcal{T}_t) + \left(B_{k'} + B_{\lfloor k'/2 \rfloor}\right) \Pr(\mathcal{E}_2 \mid \mathcal{T}_t) + B_{\lfloor k'/2 \rfloor} \Pr(\mathcal{E}_3 \mid \mathcal{T}_t) \\ &= \Big((1/2 + \Pr(\mathcal{E}_1 \mid \mathcal{T}_t) + \Pr(\mathcal{E}_2 \mid \mathcal{T}_t)\Big) B_{k'} + \Big(\Pr(\mathcal{E}_2 \mid \mathcal{T}_t) + \Pr(\mathcal{E}_3 \mid \mathcal{T}_t)\Big) B_{\lfloor k'/2 \rfloor}.\end{aligned}$$

Since $1/2 + \Pr(\mathcal{E}_1 \mid \mathcal{T}_t) + \Pr(\mathcal{E}_2 \mid \mathcal{T}_t) + \Pr(\mathcal{E}_3 \mid \mathcal{T}_t) = 1$, we have

$$B_{k'} = \mathbb{E}[N_c^u(\mathcal{T}) \mid \mathcal{T}_t] \leq \Big((1 - \Pr(\mathcal{E}_3 \mid \mathcal{T}_t)\Big) B_{k'} + \Big(\Pr(\mathcal{E}_2 \mid \mathcal{T}_t) + \Pr(\mathcal{E}_3 \mid \mathcal{T}_t)\Big) B_{\lfloor k'/2 \rfloor}.$$

Thus,

$$B_{k'} \leq \frac{\Pr(\mathcal{E}_2 \cup \mathcal{E}_3 \mid \mathcal{T}_t)}{\Pr(\mathcal{E}_3 \mid \mathcal{T}_t)} B_{\lfloor k'/2 \rfloor}.$$

Compute $\Pr(\mathcal{E}_2 \cup \mathcal{E}_3 \mid \mathcal{T}_t)$ and $\Pr(\mathcal{E}_3 \mid \mathcal{T}_t)$:

$$\Pr(\mathcal{E}_2 \cup \mathcal{E}_3 \mid \mathcal{T}_t) = \frac{1}{2d} \sum_{i=1}^{d} \Pr\left(|c_i - m_i^t| \geq (1-\varepsilon)\sqrt{\theta_t} R_t\right) = \frac{1}{2d} \sum_{i=1}^{d} \frac{(c_i - m_i^t)^2}{(1-\varepsilon)^2 R_t^2};$$
$$\Pr(\mathcal{E}_3 \mid \mathcal{T}_t) = \frac{1}{2d} \sum_{i=1}^{d} \Pr\left(|c_i - m_i^t| \geq (1+\varepsilon)\sqrt{\theta_t} R_t\right) = \frac{1}{2d} \sum_{i=1}^{d} \frac{(c_i - m_i^t)^2}{(1+\varepsilon)^2 R_t^2}.$$

Therefore, we have

$$B_{k'} \leq B_{\lfloor k'/2 \rfloor} \cdot \sum_{i=1}^{d} \frac{(c_i - m_i^t)^2}{(1-\varepsilon)^2 R_t^2} \Bigg/ \sum_{i=1}^{d} \frac{(c_i - m_i^t)^2}{(1+\varepsilon)^2 R_t^2} = \frac{(1+\varepsilon)^2}{(1-\varepsilon)^2} B_{\lfloor k'/2 \rfloor} \leq (1+5\varepsilon) B_{\lfloor k'/2 \rfloor}.$$

where the last inequality holds because $\varepsilon \leq 1/10$. □

We now bound the expected number of leaves in the threshold tree $\mathcal{T}$. By Lemma 4.5, the expected number of leaves containing center c in the threshold tree $\mathcal{T}$ is at most

$$\mathbb{E}[N_c^r(\mathcal{T})] \leq B_k \leq (1+5\varepsilon)^{\lfloor \log_2 k \rfloor} \cdot B_1 \leq \left(1 + \frac{\delta}{3 \ln k}\right)^{\log_2 k} \leq \left(e^{\frac{\delta}{3 \ln k}}\right)^{\log_2 k} < e^{\delta/2}.$$

Since $e^{\delta/2} < 1 + \delta$ for $\delta \in (0, 1)$, we have for $\delta \in (0, 1)$

$$\mathbb{E}[N_c^r(\mathcal{T})] \leq e^{\delta/2} \leq 1 + \delta.$$

□

4.4 Approximation Factor

We now prove Theorem 4.3. Our proof follows the outline given in Section 4.2. We fix a point x, step t^*, and estimate $\mathbb{E}[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_{t^*}]$. Let $c^* = \mathcal{T}_{t^*}(x)$ be the tentative center assigned to x at step t^*. As in Section 4.2, let u_t be the leaf node of $\mathcal{T}_t$ that contains x, $C_t = C_{u_t}$, $R_t = R_{u_t}$, and $m^t = m^{u_t}$. We denote the diameter of C_t by D_t.

4.4.1 Bounds on the Diameter

We prove several facts about the diameter D_t. First, we show that $D_t \approx R_t$.

Lemma 4.6. *For every leaf node u in a partially built tree $\mathcal{T}_t$, we have*

$$1/\sqrt{2} R_u \le D_u \le 2R_u.$$

Proof. The second bound easily follows from the triangle inequality: for every c' and c'' in C_u,

$$\|c' - c''\|_2 \le \|c' - m^u\|_2 + \|m^u - c''\|_2 \le 2R_u.$$

We now show the first bound. Let c be the farthest center in C_u from m^u. Then, $R_u = \|c - m^u\|_2$. Consider a center c' in C_u. The distance between c and c' is upper bounded by D_u because D_u is the diameter of C_u. Hence, for each c' in C_u, we have $\|c - c'\|_2^2 \le D_u^2$. Thus,

$$D_u^2 \ge \mathrm{Avg}_{c' \in C_u} \|c - c'\|_2^2 = \mathrm{Avg}_{c' \in C_u} \sum_{i=1}^{d} |c_i - c'_i|^2 = \sum_{i=1}^{d} \mathrm{Avg}_{c' \in C_u} |c_i - c'_i|^2,$$

where $\mathrm{Avg}_{c' \in C_u} f(c')$ denotes the average of f over c' in C_u. Observe that

$$\mathrm{Avg}_{c' \in C_u} |c_i - c'_i|^2 \geq 1/2 |c_i - m_i^u|^2.$$

This is because m^u is the median point in C_u, consequently, at least a half of all points $c' \in C_u$ are on the other side of the hyperplane $\{x : x_i = m_i^u\}$ from c (including centers c' on the hyperplane). For these centers c', we have $|c_i - c'_i| \geq |c_i - m_i^u|$. Therefore,

$$D_u^2 \geq \sum_{i=1}^{d} \mathrm{Avg}_{c' \in C_u} |c_i - c'_i|^2 \geq 1/2 \sum_{i=1}^{d} |c_i - m_i^u|^2 = 1/2 R_u^2.$$

□

We prove that the diameter D_t is exponentially decaying with t. To this end, we estimate the probability that two centers c' and c'' with $\|c' - c''\|_2 \geq D_t/2$ are separated at step t. We say that two centers $c', c'' \in C_t$ are separated at step t if $c' \notin C_t$ or $c'' \notin C_t$.

Lemma 4.7. *For every two centers $c', c'' \in C_t$ at distance at least $D_t/2$,*

$$\Pr\left\{ c' \notin C_{t+1} \text{ or } c'' \notin C_{t+1} \mid \mathcal{T}_t \right\} \geq 1/128d.$$

Proof. Suppose, at step t, the algorithm picks coordinate $i_t = i$. For every two centers $c', c'' \in C_t$, we consider the following two cases: (1) c' and c'' are on the same side of the median m^t in coordinate i (i.e. $\mathrm{sign}(c'_i - m_i^t) = \mathrm{sign}(c''_i - m_i^t)$), and (2) c' and c'' are on the opposite sides of the median m^t in coordinate i (i.e. $\mathrm{sign}(c'_i - m_i^t) = -\mathrm{sign}(c''_i - m_i^t)$).

Consider the first case, when c' and c'' are on the same side of the median m^t in coordinate i. Without loss of generality, assume that $c''_i \geq c'_i \geq m_i^t$. Observe that if $\sigma_t = 1$, $c''_i - m_i^t > (1+\varepsilon) R_t \sqrt{\theta_t}$, and $c'_i - m_i^t \leq (1-\varepsilon) R_t \sqrt{\theta_t}$, then centers c' and c'' are separated at step t. Let

$\mathcal{E}_{t,i,c'} = \{i_t = i, \sigma_t = 1\}$ be the event that the threshold cut at step t is in coordinate i and $\sigma_t = 1$. Then, the conditional probability that c' and c'' are separated given $\mathcal{E}_{t,i,c'}$ is

$$\begin{aligned}\Pr\left[c'_i - m_i^t \le (1-\varepsilon)R_t\sqrt{\theta_t} \;\&\; c''_i - m_i^t > (1+\varepsilon)R_t\sqrt{\theta_t} \mid \mathcal{T}_t,\ \mathcal{E}_{t,i,c'}\right] \\ = \Pr\left\{\theta_t \in \left[\frac{(c'_i - m_i^t)^2}{(1-\varepsilon)^2 R_t^2}, \frac{(c''_i - m_i^t)^2}{(1+\varepsilon)^2 R_t^2}\right]\right\} \\ = \left(\frac{(c''_i - m_i^t)^2}{(1+\varepsilon)^2 R_t^2} - \frac{(c'_i - m_i^t)^2}{(1-\varepsilon)^2 R_t^2}\right)^+,\end{aligned}$$

where $(x)^+$ denotes $\max\{x, 0\}$.

Now, consider the second case, when c' and c'' are on the opposite sides of the median m^u in coordinate i. Assume without loss of generality that $c''_i \ge m_i^t \ge c'_i$ and $|c''_i - m_i^t| \ge |c'_i - m_i^t|$. If $c''_i - m_i^u \ge (1+\varepsilon)R_t\sqrt{\theta_t}$ and $\sigma_t = 1$, then c' and c'' are separated at this step. Thus, the conditional probability that c' and c'' are separated given $i_t = i$ and parameter $\sigma_t = 1$ is at least

$$\Pr\left(c''_i - m_i^t \ge (1+\varepsilon)R_t\sqrt{\theta_t} \mid \mathcal{T}_t, i_t = i, \sigma_t = 1\right) = \frac{(c''_i - m_i^t)^2}{(1+\varepsilon)^2 R_t^2}.$$

Define

$$a_i = \min\left\{|c'_i - m_i^t|, |c''_i - m_i^t|\right\} \quad \text{and} \quad b_i = \max\left\{|c'_i - m_i^t|, |c''_i - m_i^t|\right\}.$$

Let $I_1, I_2 \subset \{1, 2, \ldots, d\}$ be the set of indices i for which c'_i and c''_i lie on the same side and

opposite sides of m^t, respectively. Then,

$$\Pr\Big\{c' \notin C_{t+1} \text{ or } c'' \notin C_{t+1} \mid \mathcal{T}_t\Big\} \geq \\ \geq \frac{1}{2d}\sum_{i\in I_1}\Big(\frac{b_i^2}{(1+\varepsilon)^2R_t^2} - \frac{a_i^2}{(1-\varepsilon)^2R_t^2}\Big)^+ + \frac{1}{2d}\sum_{i\in I_2}\frac{b_i^2}{(1+\varepsilon)^2R_t^2}.$$

Now observe that

$$\frac{1}{2d}\sum_{i\in I_1}\Big(\frac{b_i^2}{(1+\varepsilon)^2R_t^2} - \frac{a_i^2}{(1-\varepsilon)^2R_t^2}\Big)^+ \geq \\ \geq \frac{1}{2dR_t^2}\sum_{i\in I_1}\frac{b_i^2}{(1+\varepsilon)^2} - \frac{a_i^2}{(1-\varepsilon)^2} \\ \geq \frac{1}{2dR_t^2}\sum_{i\in I_1} b_i^2 - a_i^2 - (2\varepsilon b_i^2 + 3\varepsilon a_i^2).$$

Similarly, we have

$$\frac{1}{2d}\sum_{i\in I_2}\frac{b_i^2}{(1+\varepsilon)^2R_t^2} \geq \frac{\sum_{i\in I_2} b_i^2 - 2\varepsilon b_i^2}{2R_t^2 d}.$$

When c' and c'' are on the same side of m^t in coordinate i, we have

$$b_i^2 - a_i^2 = (b_i - a_i)(b_i + a_i) \geq (b_i - a_i)^2 = (c_i' - c_i'')^2.$$

When c' and c'' are on the opposite side of m^t in coordinate i, we have

$$4b_i^2 \geq (b_i + a_i)^2 = (c_i' - c_i'')^2.$$

Note that $\sum_{i=1}^d b_i^2 + a_i^2 = \sum_{i=1}^d (c_i' - m_i^t)^2 + (c_i'' - m_i^t)^2 = \|c' - m^t\|_2^2 + \|c'' - m^t\|_2^2$. Therefore,

the probability that c' and c'' are separated at step t is at least

$$\begin{aligned}\Pr\left(c' \notin C_{t+1} \text{ or } c'' \notin C_{t+1} \mid \mathcal{T}_t\right) &\geq \sum_{i=1}^{d} \frac{(c'_i - c''_i)^2}{8dR_t^2} - \frac{2\varepsilon b_i^2 + 3\varepsilon a_i^2}{2dR_t^2} \\ &\geq \frac{\|c' - c''\|_2^2}{8R_t^2 d} - \frac{6\varepsilon}{2d}.\end{aligned}$$

where the second inequality is due to $\sum_{i=1}^{d} 2\varepsilon b_i^2 + 3\varepsilon a_i^2 \leq \sum_{i=1}^{d} 3\varepsilon b_i^2 + 3\varepsilon a_i^2 \leq 3\varepsilon\|c' - m^t\|_2^2 + 3\varepsilon\|c'' - m^t\|_2^2 \leq 6\varepsilon R_t^2$. We conclude that for centers c' and c'' with $\|c' - c''\|_2^2 \geq D_t^2/4$, we have

$$\begin{aligned}\Pr\left(c' \notin C_{t+1} \text{ or } c'' \notin C_{t+1} \mid \mathcal{T}_t\right) &\geq \frac{1}{2d} \cdot \left(\frac{D_t^2}{16R_t^2} - 5\varepsilon\right) \\ &\geq \frac{1}{2d} \cdot \left(\frac{1}{32} - 5\varepsilon\right) \geq \frac{1}{128d}.\end{aligned}$$

Here, we used that $D_t \geq {}^1/\sqrt{2}R_t$ and $\varepsilon \leq {}^1/_{384}$. □

We obtain the following corollary from Lemma 4.6.

Lemma 4.8. *Let $L = \lceil 640d \ln k \rceil$. Then, for every t, we have*

$$\Pr\left(D_{t+L} \geq D_t/2 \mid \mathcal{T}_t\right) \leq \frac{1}{k^3}.$$

Proof. Consider a fixed time step t. Suppose the distance between centers c' and c'' is at least $D_t/2$. Since the diameter D_t is non-increasing as t increases, the distance between c' and c'' is greater than $D_{t'}/2$ for any step $t' \geq t$. By Lemma 4.7, the probability that these centers c' and c'' are separated at step t' is at least ${}^1/_{128d}$.

Thus, these two centers c' and c'' are not separated in $\lceil 640d \ln k \rceil$ steps from step t with probability at most

$$\left(1 - \frac{1}{128d}\right)^{640d \ln k} \leq e^{-5 \ln k}.$$

Since there are at most $\binom{k}{2}$ pairs of centers with distance greater than $D_t/2$, by the union bound over all such pairs, we have for $L = \lceil 640d \ln k \rceil$

$$\Pr\left(D_{t+L} \geq D_t/2 \mid \mathcal{T}_t\right) \leq \binom{k}{2} \cdot e^{-5 \ln k} \leq \frac{1}{k^3}.$$

□

To simplify the exposition, we define a stopping time t^{**}. Let t^{**} be the first step $t > t^*$ of the algorithm when one of the following happens: (A) $D_t \leq \|x - c^*\|_2$ (note: if c^* is the only center remaining in C_t, then $D_t = 0$); (B) x and c^* are separated before step t (i.e., $c^* \notin C_t$); or (C) $D_t > D_{t-L'}/2$ and $t \geq t^* + L'$ for $L' = \lceil 1280d \ln k \rceil$. For some step t, C_t contains only one center and $D_t = 0$. Thus, the stopping time t^{**} is well-defined. We show that it is very unlikely that the case (C) happens, i.e. $D_{t^{**}} > D_{t^{**}-L'}/2$ and $t^{**} \geq t^* + L'$.

Corollary 4.9. *Let $L' = \lceil 1280d \ln k \rceil$ be twice as large as L in Lemma 4.8. Then,*

$$\Pr\left(D_{t^{**}} > D_{t^{**}-L'}/2 \;\&\; t^{**} \geq t^* + L' \mid \mathcal{T}_{t^*}\right) \leq \frac{1}{k}.$$

Proof. Let $L = \lceil 640d \ln k \rceil$ be as in Lemma 4.8. We consider the set of steps

$$S_L = \{t \leq t^{**} : t = t^* + Lz, z \geq 1\}.$$

By Lemma 4.8, we have for each step $t = t^* + Lz$ in this set S_L

$$\Pr\left(D_t > D_{t-L}/2 \mid \mathcal{T}_{t-L}\right) \leq \frac{1}{k^3}.$$

We consider every step $t = t^* + L'z$ for $z \geq 1$. If $D_t > D_{t-L'}/2$, then we have $t^{**} \leq t$. If

$D_t \leq D_{t-L'}/2$, then we must separate at least one center from $C_{t-L'}$ in L' steps, which means $|C_t| < |C_{t-L'}|$. Since there are at most k centers in C_{t^*}, we have at most k such steps t with $D_t \leq D_{t-L'}/2$. Thus, we have $t^{**} \leq t^* + L'k = t^* + 2kL$. Then, the set of steps S_L contains at most $2k$ steps. By the union bound over all steps $t \in S_L$, we have $D_t \leq D_{t-L}/2$ for all steps $t \in S_L$ with probability at least $1 - 1/k$. Suppose that $D_t \leq D_{t-L}/2$ holds for all steps $t \in S_L$. For every $t^* + L' \leq t \leq t^{**}$, there exists a $t' \in S_L$ such that $t - L' \leq t' - L < t' \leq t$. Since D_t is a non-increasing sequence, we have for every $t^* + L' \leq t \leq t^{**}$

$$D_t \leq D_{t'} \leq D_{t'-L}/2 \leq D_{t-L'}/2.$$

Therefore, we have $D_{t^{**}} > D_{t^{**}-L'}/2$ and $t^{**} \geq t^* + L'$ with probability at most $1/k$. □

4.4.2 Cost of Separation

In this section, we complete the proof of Theorem 4.3. The proof is similar to the overview we gave in Section 4.2. The key difference is that we no longer assume that the distance from x to the nearest fallback center does not depend on the cut that separates x and c^*.

To simplify the exposition, from now on, we shall assume that $c_i^* \geq x_i$ for all i. We make this assumption without loss of generality, because if $c_i^* < x_i$ for some i, we can mirror all centers c in C and point x across the hyperplane $\{y_i = 0\}$, or, in other words, we can change the sign of the i-th coordinate for all centers c in C and point x. This transformation does not affect the algorithm but makes $c_i^* \geq x_i$.

For every (i, η) with $x_i \leq \eta < c_i$, define $M_t(i, \eta)$ as follows: $M_t(i, \eta)$ equals the distance from x to the closest center c' in C_t with $c'_i \leq \eta$. If there are no centers c' in C_t with $c'_i \leq \eta$, then we let

$M_t(i,\eta) = \infty$. Observe that if x and c^* are separated at step t, then

$$x_i \le m_i^t + \sigma_t\sqrt{\theta_t}R_t < \underbrace{m_i^t + \sigma_t\sqrt{\theta_t}R_t + \varepsilon\sqrt{\theta_t}R_t}_{\eta_t} < c_i^*,$$

where i is the coordinate chosen at step t. Thus, if x and c^* are separated at step t, the distance from x to the fallback center is $M_t(i,\eta_t)$, where $\eta_t = m_i^t + \sigma_t\sqrt{\theta_t}R_t + \varepsilon\sqrt{\theta_t}R_t$.

At each step t, our algorithm calls function *Divide-and-Share* with parameters $(i_t, \sigma_t, \theta_t)$ to split node u_t. Let $\omega_t = (i_t, \xi_t)$ be the cut chosen by the algorithm for node u_t where $\xi_t = m_i^t + \sigma_t\sqrt{\theta_t}R_t$; ω_t is undefined ($\omega_t = \perp$), if the algorithm does not make any cut at step t. Note that the cut ω_t is determined by the tuple $(i_t, \sigma_t, \theta_t)$. Then, x and c^* are separated at step t by the tuple (i, σ, θ) if $c^* \in C_t$, $\omega_t = (i, m_i^t + \sigma\sqrt{\theta}R_t)$ and $x_i \le \xi_t < \eta_t < c_i^*$.

We define a penalty function $Z_t(i,\sigma,\theta)$ for every tuple (i,σ,θ) with $i \in \{1, 2 \ldots, d\}, \sigma \in \{\pm 1\}, \theta \in (0,1)$ as follows:

$$Z_t(i,\sigma,\theta) = \begin{cases} \mathbb{E}\big[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_t,\ \omega_t = (i, m_i^t + \sigma\sqrt{\theta}R_t)\big], & \text{if } (i,\sigma,\theta) \text{ separates } x \ \&\ c^* \text{ at step } t; \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $Z_t(i,\sigma,\theta)$ equals 0 if the tuple (i,σ,θ) does not separate x and c^* at step t. Otherwise, it is equal to the expected cost of x in the final tree $\mathcal{T}$ assuming that the algorithm chooses the tuple (i,σ,θ) at step t. Note that if x and c^* are already separated at step t, then $Z_t(i,\sigma,\theta) = 0$.

Claim 4.10. *For every step t and every tuple (i,σ,θ), we have*

$$Z_t(i,\sigma,\theta) \le \min\big\{2\|x - c^*\|_2^2 + 2D_t^2, A_k\ M_t^2(i,\eta)\big\},$$

where $\eta = m_i^t + (\sigma + \varepsilon)\sqrt{\theta}R_t$.

Proof. If x and c^* are not separated by the tuple (i, σ, θ) at step t or x and c^* are already separated at step t, then we have $Z_t(i, \sigma, \theta) = 0$. Thus, we only need to consider the case when x and c^* are separated by the tuple (i, σ, θ) at step t. By the triangle inequality, we have

$$\|x - \mathcal{T}(x)\|_2^2 \leq (\|x - c^*\|_2 + \|c^* - \mathcal{T}(x)\|_2)^2 \leq (\|x - c^*\|_2 + D_t)^2 \leq 2\|x - c^*\|_2 + 2D_t^2.$$

By Definition 4.2.2 of the approximation factor A_k, we have

$$Z_t(i, \sigma, \theta) = \mathbb{E}\Big[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_t,\ \omega_t = (i, m_i^t + \sigma\sqrt{\theta}R_t)\Big] \leq A_k\|x - \mathcal{T}_{t+1}(x)\|_2^2 = A_k M_t^2(i, \eta).$$

Combining these two bounds, we get the conclusion. □

Our goal is to show that $A_k \leq O(1/\varepsilon \log k \log\log k)$. We prove Lemma 4.11, which provides the following recurrence relation on A_k: $A_k \leq \max\{4, A_k/k\} + \alpha/\varepsilon \log k \log A_k$. Using this recurrence relation, we get the desired bound on A_k.

Lemma 4.11. *For some absolute constant α, we have*

$$\mathbb{E}\left[\frac{\|x - \mathcal{T}(x)\|_2^2}{\|x - c^*\|_2^2} \mid \mathcal{T}_{t^*}\right] \leq \max\{4, A_k/k\} + \alpha/\varepsilon \log k \log A_k. \tag{4.9}$$

Proof. Let t^{**} be the stopping time from Corollary 4.9: t^{**} is the first step t when (A) $D_t \leq \|x - c^*\|_2$ (note: if c^* is the only center remaining in C_t, then $D_t = 0$); (B) x and c^* are separated before step t (i.e., $c^* \notin C_t$); or (C) $D_t > D_{t-L'}/2$ (where $L' = O(d \ln k)$ as in Corollary 4.9; $t \geq t^* + L'$). Let $\mathcal{E}_A$, $\mathcal{E}_B$, and $\mathcal{E}_C$ be events corresponding to the the stopping rules (A), (B), and

(C):

$$\mathcal{E}_A = \left\{D_{t^{**}} \le \|x - c^*\|_2 \ \&\ c^* \in C_{t^{**}}\right\};$$
$$\mathcal{E}_B = \left\{x \ \&\ c^* \text{ are separated at step } t^{**} - 1\right\};$$
$$\mathcal{E}_C = \left\{D_{t^{**}} > D_{t^{**}-L'}/2 \ \&\ t^{**} \ge t^* + L'\right\} \setminus (\mathcal{E}_A \cup \mathcal{E}_B).$$

Note that $\mathcal{E}_A$, $\mathcal{E}_B$, and $\mathcal{E}_C$ are disjoint collectively exhaustive events (one of them must always occur) and by Corollary 4.9, $\Pr(\mathcal{E}_C \mid \mathcal{T}_{t^*}) \le 1/k$. We further partition $\mathcal{E}_B$ into disjoint events

$$\mathcal{E}_{B,t} = \{x \ \&\ c^* \text{ are separated at step } t\}.$$

If event $\mathcal{E}_A$ occurs, then the eventual cost of x is at most $(\|x - c^*\|_2 + D_{t^{**}})^2 \le 4\|x - c^*\|_2^2$ because every center in $C_{t^{**}}$ is at distance at most $\|x - c^*\|_2 + D_{t^{**}}$ from x. If event $\mathcal{E}_{B,t}$ occurs, then the expected cost of x is upper bounded by $Z(i_t, \sigma_t, \theta_t)$. Finally, if event $\mathcal{E}_C$ occurs, then the expected cost of x in $\mathcal{T}$ is upper bounded by $A_k\|x - c^*\|_2^2$ (because c^* is the tentative center for x at step t^{**}). We have

$$\begin{aligned}\mathbb{E}\left[\|x - \mathcal{T}(x)\|_2^2 \mid \mathcal{T}_{t^*}\right] &\le 4\|x - c^*\|_2^2 \cdot \Pr(\mathcal{E}_A \mid \mathcal{T}_{t^*}) + A_k\|x - c^*\|_2^2 \cdot \Pr(\mathcal{E}_C \mid \mathcal{T}_{t^*}) \\ &\quad + \sum_{t=t^*}^{\infty} \mathbb{E}\Big[Z_t(i_t, \sigma_t, \theta_t) \mid \mathcal{E}_{B,t},\ \mathcal{T}_{t^*}\Big] \Pr(\mathcal{E}_{B,t} \mid \mathcal{T}_{t^*}) \\ &\le \max\{4, A_k/k\} \cdot \|x - c^*\|_2^2 + \sum_{t=t^*}^{\infty} \mathbb{E}\Big[(Z_t(i_t, \sigma_t, \theta_t) - 4\|x - c^*\|_2^2) \cdot \mathbf{1}(\mathcal{E}_{B,t}) \mid \mathcal{T}_{t^*}\Big].\end{aligned}$$

Let $\widetilde{Z}_t(i_t, \sigma_t, \theta_t) = \max\{Z_t(i_t, \sigma_t, \theta_t) - 4\|x - c^*\|_2^2, 0\}$. Then,

$$\mathbb{E}\left[\frac{\|x - \mathcal{T}(x)\|_2^2}{\|x - c^*\|_2^2} \mid \mathcal{T}_{t^*}\right] \le \max\{4, {}^{A_k}/_k\} + \sum_{t=t^*}^{\infty} \mathbb{E}\left[\frac{\widetilde{Z}_t(i_t, \sigma_t, \theta_t)}{\|x - c^*\|_2^2} \cdot \mathbf{1}(\mathcal{E}_{B,t}) \mid \mathcal{T}_{t^*}\right].$$

Our goal is to upper bound the second term by $\alpha/\varepsilon \log k \log A_k$. Write,

$$\mathbb{E}\left[\widetilde{Z}_t(i_t,\sigma_t,\theta_t)\cdot\mathbf{1}(\mathcal{E}_{B,t}) \mid \mathcal{T}_{t^*}\right] = \sum_{i=1}^{d}\mathbb{E}\left[\int_0^1 \frac{\widetilde{Z}_t(i,-1,\theta)+\widetilde{Z}_t(i,1,\theta)}{2d}\,d\theta\cdot\mathbf{1}\{t<t^{**}\} \mid \mathcal{T}_{t^*}\right]. \tag{4.10}$$

Here, we used that parameters i_t, σ_t, and θ_t are randomly chosen from $\{1,\dots,d\}$, $\{\pm 1\}$, and $[0,1]$, respectively. We need the following lemma, which we prove in Section 4.4.3.

Lemma 4.12. *For every i, we have*

$$\int_0^1 \frac{\widetilde{Z}_t(i,-1,\theta)+\widetilde{Z}_t(i,1,\theta)}{2}\,d\theta \le \frac{c_i^*-x_i}{\varepsilon(1-\varepsilon)}\int_{x_i}^{c_i^*}\frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2}\,d\eta.$$

Using Lemma 4.12, we can upper bound (4.10) as follows

$$\begin{aligned}
\mathbb{E}\left[\widetilde{Z}_t(i_t,\sigma_t,\theta_t)\cdot\mathbf{1}(\mathcal{E}_{B,t}) \mid \mathcal{T}_{t^*}\right] &\le \frac{1}{d}\sum_{i=1}^{d}\frac{c_i^*-x_i}{\varepsilon(1-\varepsilon)}\,\mathbb{E}\left[\sum_{t=t^*}^{t^{**}-1}\int_{x_i}^{c_i^*}\frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2}\,d\eta \mid \mathcal{T}_{t^*}\right] \\
&= \frac{1}{d}\sum_{i=1}^{d}\frac{c_i^*-x_i}{\varepsilon(1-\varepsilon)}\int_{x_i}^{c_i^*}\mathbb{E}\left[\sum_{t=t^*}^{t^{**}-1}\frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2} \mid \mathcal{T}_{t^*}\right]d\eta \\
&\le \frac{1}{d}\sum_{i=1}^{d}\frac{2(c_i^*-x_i)^2}{\varepsilon}\max_{\eta\in[x_i,c_i^*]}\mathbb{E}\left[\sum_{t=t^*}^{t^{**}-1}\frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2} \mid \mathcal{T}_{t^*}\right].
\end{aligned}$$

We now show that for every $\eta\in[x_i,c_i^*]$ the following bound holds with probability 1:

$$\sum_{t=t^*}^{t^{**}-1}\frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2} \le O(d\log k\log A_k). \tag{4.11}$$

This will conclude the proof of Lemma 4.11 because (4.11) implies that

$$\mathbb{E}\left[\widetilde{Z}_t(i_t,\sigma_t,\theta_t)\cdot\mathbf{1}(\mathcal{E}_{B,t}) \mid \mathcal{T}_{t^*}\right] \le \frac{1}{d}\sum_{i=1}^{d}\frac{2(c_i^*-x_i)^2}{\varepsilon}\cdot O(d\log k\log A_k) = \frac{2\|c^*-x\|_2^2}{\varepsilon}\cdot O(\log k\log A_k).$$

□

Lemma 4.13. *Inequality (4.11) holds with probability* 1.

Proof. By Lemma 4.6, $R_t \geq D_t/2$. Thus,

$$\sum_{t=t^*}^{t^{**}-1} \frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2} \leq 8 \sum_{t=t^*}^{t^{**}-1} \frac{\min\{D_t^2, A_k M_t^2(i,\eta)\}}{D_t^2} = 8 \sum_{t=t^*}^{t^{**}-1} \min\left\{1, \frac{A_k\, M_t^2(i,\eta)}{D_t^2}\right\}.$$

Let

$$f_t(i,\eta) = \frac{A_k\, M_t^2(i,\eta)}{D_t^2}.$$

Observe that $M_t(i,\eta)$ is a non-decreasing sequence and D_t is a non-increasing sequence for fixed i, η and $t \in \{t^*, \ldots, t^{**}-1\}$. Moreover, by the definition of stopping time t^{**}, $D_t \leq D_{t-L'}/2$ for $t \in \{t^*+L', \ldots, t^{**}-1\}$, where $L' = O(d \log k)$ (see stopping rule (C)). Hence, $f_t(i,\eta)$ is a non-decreasing sequence, and $f_t(i,\eta) \geq 4 f_{t-L'}(i,\eta)$ for $t \in \{t^*+L', \cdots, t^{**}-1\}$. Let t' be the first step t in $[t^*, t^{**}-1]$ when $f_{t'}(i,\eta) \geq 1$. If $f_t(i,\eta) < 1$ for all $t \in \{t^*, \cdots, t^{**}-1\}$, then $t' = t^{**}$. We have

$$\frac{1}{8} \sum_{t=t^*}^{t^{**}-1} \frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2} \leq \sum_{t=t^*}^{t^{**}-1} \min\{1, f_t(i,\eta)\} = \underbrace{\sum_{t=t^*}^{t'-1} f_t(i,\eta)}_{\Sigma_I} + \underbrace{\sum_{t=t'}^{t^{**}-1} 1}_{\Sigma_{II}}.$$

The first sum (Σ_I) on the right hand side is upper bounded by $2L' \cdot f_{t'}(i,\eta)$, because $f_t(i,\eta) \geq 4 f_{t-L'}(i,\eta)$ for $t < t^{**}$. In turn, $2L' \cdot f_{t'}(i,\eta) \leq 2L' = O(d \log k)$, because $f_t(i,\eta) \leq 1$ for $t < t'$. The second sum (Σ_{II}) equals $t^{**} - t'$. Since $f_t(i,\eta) \geq 4 f_{t-L'}(i,\eta)$ for every $t \in [t^*+L', t^{**}-1]$, we have

$$\left\lfloor \frac{(t^{**}-1) - t'}{L'} \right\rfloor \leq \log_4 \frac{f_{t^{**}-1}(i,\eta)}{f_{t'}(i,\eta)} \leq \log_4 f_{t^{**}-1}(i,\eta) = \log_4 \left(\frac{A_k\, M_{t^{**}-1}^2(i,\eta)}{D_{t^{**}-1}^2} \right).$$

It remains to show that $M_{t^{**}-1}(i,\eta) = O(D_{t^{**}-1})$ and thus

$$t^{**} - t' = O(L' \log A_k) = O(d \log k \log A_k).$$

We have, $M_{t^{**}-1}(i,\eta) \leq \|x - c^*\|_2 + D_t \leq 2D_t$, where we used that for every $t < t^{**}$, $D_t > \|x - c^*\|_2$ (see stopping rule (C)). This finishes the proof of Lemma 4.13. □

4.4.3 Proof of Lemma 4.12

We first make the following simple but crucial observation.

Claim 4.14. *If $\widetilde{Z}_t(i,\sigma,\theta) > 0$, then for $\eta = m_i^t + (\sigma+\varepsilon)\sqrt{\theta}R_t$, we have*

$$|\eta - m_i^t| \equiv |(\sigma+\varepsilon)\sqrt{\theta}R_t| \leq \frac{c_i^* - x_i}{\varepsilon}.$$

Proof of Claim 4.14. If $\widetilde{Z}_t(i,\sigma,\theta) > 0$, then the cut with parameters i, σ, θ separates x and c^* (otherwise, $Z_t(i,\sigma,\theta)$ and $\widetilde{Z}_t(i,\sigma,\theta)$ would be equal to 0). That is, $x_i \leq m_i^t + \sigma\sqrt{\theta}R_t$ and $c_i^* > m_i^t + (\sigma+\varepsilon)\sqrt{\theta}R_t$. Write,

$$c_i^* - x_i = (c_i^* - m_i^t) - (x_i - m_i^t) > (\sigma+\varepsilon)\sqrt{\theta}R_t - \sigma\sqrt{\theta}R_t = \varepsilon\sqrt{\theta}R_t.$$

Hence,

$$|(\sigma+\varepsilon)\sqrt{\theta}R_t| = \frac{|\sigma+\varepsilon|}{\varepsilon} \cdot \varepsilon\sqrt{\theta}R_t < \frac{|\sigma+\varepsilon|}{\varepsilon}(c_i^* - x_i).$$

□

Proof of Lemma 4.12. We have

$$\int_0^1 \frac{\widetilde{Z}_t(i,-1,\theta)+\widetilde{Z}_t(i,1,\theta)}{2}\,d\theta = \frac{1}{2}\sum_{\sigma\in\{\pm 1\}}\int_0^1 \widetilde{Z}_t(i,\sigma,\theta)\,d\theta.$$

Make the substitutions $\eta_\sigma = m_i^t + (\sigma+\varepsilon)R_t\sqrt{\theta}$. Then, $d\theta = \frac{2(\eta_\sigma - m_i^t)}{(\sigma+\varepsilon)^2 R_t^2} d\eta_\sigma$ and

$$\int_0^1 \frac{\widetilde{Z}_t(i,-1,\theta)+\widetilde{Z}_t(i,1,\theta)}{2}\,d\theta = \sum_{\sigma\in\{\pm 1\}}\int_{m_i^t}^{m_i^t+(\sigma+\varepsilon)R_t} \frac{\widetilde{Z}_t(i,\sigma,\theta)}{(\sigma+\varepsilon)^2 R_t^2}\cdot(\eta_\sigma - m_i^t)\,d\eta_\sigma.$$

By Claim 4.12, $|\eta_\sigma - m_i^t| \le |\sigma+\varepsilon|/\varepsilon\cdot(c_i^* - x_i)$. Since $Z(i,\sigma,\theta)\ge 0$, we have $\widetilde{Z}(i,\sigma,\theta) = \max\{Z_t(i_t,\sigma_t,\theta_t) - 4\|x-c^*\|_2^2, 0\} \le Z(i,\sigma,\theta)$. As we discuss in the previous section, $\widetilde{Z}(i,\sigma,\theta) \le Z(i,\sigma,\theta) \le \min\{2D_t^2, A_k M_t^2(i,\eta_\sigma)\}$ (see Claim 4.10). Also, if $\eta_\sigma \notin [x_i, c_i^*]$, then x and c^* are not separated by the tuple (i,σ,θ), which implies $\widetilde{Z}(i,\sigma,\theta) = 0$. Thus,

$$\int_0^1 \frac{\widetilde{Z}_t(i,-1,\theta)+\widetilde{Z}_t(i,1,\theta)}{2}\,d\theta \le \frac{c_i^* - x_i}{\varepsilon(1-\varepsilon)}\int_{x_i}^{c_i^*}\frac{\min\{2D_t^2, A_k M_t^2(i,\eta)\}}{R_t^2}\,d\eta.$$

This concludes the proof of Lemma 4.12. □

CHAPTER 5
CONCLUSION AND OPEN PROBLEMS

5.1 Conclusion

Clustering, as a fundamental task in data analysis, is widely used in business, engineering, and science. In practice, centroid-based clustering is one of the most popular clustering methods, including k-means and k-medians. Efficient algorithms like k-means++ and k-means$\|$ achieve good approximations for these clustering. However, the k-means and k-medians clustering form a Voronoi partition of the entire space, which usually has complicated boundaries. Thus, the regular k-means and k-medians clustering is not necessarily easy to understand by humans. This thesis focus on explainable clustering with k-means and k-medians objective proposed by Dasgupta et al. (2020). We design new approximation algorithms for these explainable clustering problems. We give a bi-criteria approximation algorithm for explainable k-means, which captures the tradeoff between accuracy and explainability. We hope our work will help further studies in explainable artificial intelligence(XAI), which creates comprehensible and trustworthy results and output by using machine learning algorithms.

5.2 Open Problems

We list below several interesting open problems.

Better Bounds on the Price of Explainability: For explainable k-medians in ℓ_1, we show that the RANDOMCOORDINATECUT algorithm achieves the tight competitive ratio. An open problem is whether we can improve the bounds on the Price of Explainability for k-means and k-medians

in ℓ_2 and also the bounds for bi-criteria explainable k-means. Gupta et al. (2023) improved the competitive ratio for explainable k-means from $O(k \log k)$ to $O(k \log \log k)$. The lower bound of the price of explainability for k-means is $\Omega(k)$. The upper bound and lower bound of the price of explainability for k-medians in ℓ_2 are $O(\log^{3/2} k)$ and $\Omega(k)$ respectively.

Approximation of the Optimal Explainable Clustering: Recently, Bandyapadhyay et al. (2022) and Laber (2022) proposed the following problem: Can we get better approximation ratios compared to the cost of the optimal explainable clustering instead of the unconstrained clustering? Gupta et al. (2023) showed that the explainable k-medians and k-means are hard to approximate better than $O(\log k)$ unless P=NP. An open problem is whether we can approximate the explainable k-means better than $\tilde{O}(k)$, which is the upper bound of the price of explainability for k-means.

Hierarchical Explainable Clustering: An interesting observation is that the threshold tree naturally creates hierarchical clustering. Hierarchical clustering can describe the data at many levels of granularity. It is known that there exists a hierarchical clustering for k-centers Dasgupta and Long (2005) and k-medians Plaxton (2006) such that for every k, the induced k clustering is a constant approximation to the optimal k clustering. However, the hierarchical clustering given by these algorithms is not necessarily explainable in terms of the boundaries of clusters. Thus, an open problem is to find a threshold tree to provide a good hierarchical clustering for k-medians and k-means.

Shallow Threshold Tree: The threshold decision tree is easy to understand by humans because it uses only k threshold cuts to partition the space. However, the depth of this decision tree can be $k-1$ in the worst case. The clusters corresponding to the leaves with smaller depths are easier to understand since it depends on fewer threshold cuts. Thus, Laber et al. (2023) proposed to create a shallow decision tree to describe clusters. They provided a heuristic algorithm that achieves lower or equivalent costs with considerably shallower trees compared to previous explainable clustering

algorithms by Dasgupta et al. (2020); Frost et al. (2020); Laber and Murtinho (2021). Recently, Deng et al. (2023) found an instance in $\mathbb{R}^2$, for which there exists a decision tree with depth $k-1$ achieves the same cost as the optimal unconstrained clustering, while any decision tree with $k-2$ depth has an unbound cost. An interesting problem is whether we can get a good approximation with a bi-criteria shallow threshold tree.

Well-Clusterable Instance: On real-world datasets, we observe that the greedy algorithm by Dasgupta et al. (2020) usually achieves better performance than our algorithms. The heuristic algorithm by Laber et al. (2023) finds a shallow decision tree with a shallower decision tree in practice. A natural question is whether there exist some common properties in real-world datasets such that we can achieve better approximation. Papanikolaou (2023) showed that the greedy algorithm achieves a constant competitive ratio for k-means if the instance is a-separated for some $a \geq 12k\sqrt{d}$. However, the hard instance used in the lower bound for explainable k-means is also $k\sqrt{d}$-separated. An open problem is whether there exist some other natural properties of real-world instance or other explainable notion for clustering.

REFERENCES

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

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, 49(4): FOCS17–97, 2019.

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

David Arthur and Sergei Vassilvitskii. k-means++ the advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. *arXiv preprint arXiv:1502.03316*, 2015.

Olivier Bachem, Mario Lucic, and Andreas Krause. Distributed and provably good seedings for k-means in constant rounds. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 292–300. JMLR. org, 2017.

Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.

Sayan Bandyapadhyay, Fedor Fomin, Petr A Golovach, William Lochet, Nidhi Purohit, and Kirill Simonov. How to find a good explanation for clustering? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3904–3912, 2022.

Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for k-means: beyond subspaces and the johnson-lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1039–1050, 2019.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

Dimitris Bertsimas, Agni Orfanoudaki, and Holly Wiberg. Interpretable clustering via optimal trees. *arXiv preprint arXiv:1812.00539*, 2018.

Brad Boehmke and Brandon M Greenwell. *Hands-on machine learning with R*. CRC Press, 2019.

Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 968–977. SIAM, 2009.

Christos Boutsidis, Anastasios Zouzias, Michael W Mahoney, and Petros Drineas. Randomized dimensionality reduction for k-means clustering. *IEEE Transactions on Information Theory*, 61 (2):1045–1062, 2014.

Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

Tobias Brunsch and Heiko Röglin. A bad instance for k-means++. *Theoretical Computer Science*, 505:19–26, 2013.

Jarosław 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 (TALG)*, 13(2):1–31, 2017.

Moses Charikar and Lunjia Hu. Near-optimal explainable k-means for all dimensions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2580–2606. SIAM, 2022.

Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 1–10, 1999.

Michael B Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 163–172, 2015.

Vincent Cohen-Addad and Euiwoong Lee. Johnson coverage hypothesis: Inapproximability of k-means and k-median in lp-metrics. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1493–1530. SIAM, 2022.

Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1621–1628, 2022.

Sanjoy Dasgupta. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California, San Diego, 2008.

Sanjoy Dasgupta. UCSD CSE 291, Lecture Notes: Geometric Algorithms, 2013. URL: https://cseweb.ucsd.edu/~dasgupta/291-geom/kmeans.pdf. Last visited on 2020/06/01.

Sanjoy Dasgupta and Philip M Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.

Sanjoy Dasgupta, Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Explainable k-means and k-medians clustering. In *International Conference on Machine Learning*, pages 7055–7065. PMLR, 2020.

Chengyuan Deng, Surya Teja Gavva, Parth Patel, Karthik C. S., and Adarsh Srinivasan. Impossibility of depth reduction in explainable clustering. *arXiv preprint arXiv:2305.02850*, 2023.

Dheeru Dua and Casey Graff. UCI ML repository, 2017. URL http://archive.ics.uci.edu/ml.

Ron Elber. Kdd-Cup, 2004. URL http://osmot.cs.cornell.edu/kddcup/.

Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Almost tight approximation algorithms for explainable clustering. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2641–2663. SIAM, 2022.

Ricardo Fraiman, Badih Ghattas, and Marcela Svarc. Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7(2):125–145, 2013.

Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Exkmc: Expanding explainable k-means clustering. *arXiv preprint arXiv:2006.02399*, 2020.

Buddhima Gamlath, Xinrui Jia, Adam Polak, and Ola Svensson. Nearly-tight and oblivious algorithms for explainable clustering. *Advances in Neural Information Processing Systems*, 34: 28929–28939, 2021.

Fabrizio Grandoni, Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Rakesh Venkat. A refined approximation for euclidean k-means. *Information Processing Letters*, 176:106251, 2022. ISSN 0020-0190.

Anupam Gupta, Madhusudhan Reddy Pittu, Ola Svensson, and Rachel Yuan. The price of explainability for clustering. *arXiv preprint arXiv:2304.09743*, 2023.

John Frank Charles Kingman. *Poisson processes*, volume 3. Clarendon Press, 1992.

Eduardo Laber and Lucas Murtinho. On the price of explainability for some clustering problems. In *International Conference on Machine Learning*. PMLR, 2021.

Eduardo Laber, Lucas Murtinho, and Felipe Oliveira. Shallow decision trees for explainable k-means clustering. *Pattern Recognition*, 137:109239, 2023.

Eduardo Sany Laber. The computational complexity of some explainable clustering problems. *arXiv preprint arXiv:2208.09643*, 2022.

Silvio Lattanzi and Christian Sohler. A better k-means++ algorithm via local search. In *International Conference on Machine Learning*, pages 3662–3671, 2019.

Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Information Processing Letters*, 120:40–43, 2017.

Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. In *proceedings of the forty-fifth annual ACM symposium on theory of computing*, pages 901–910, 2013.

Bing Liu, Yiyuan Xia, and Philip S Yu. Clustering via decision tree construction. In *Foundations and advances in data mining*, pages 97–124. Springer, 2005.

Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.

Konstantin Makarychev and Liren Shan. Near-optimal algorithms for explainable k-medians and k-means. In *International Conference on Machine Learning*, pages 7358–7367. PMLR, 2021.

Konstantin Makarychev and Liren Shan. Explainable k-means: don't be greedy, plant bigger trees! In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1629–1642, 2022.

Konstantin Makarychev and Liren Shan. Random cuts are optimal for explainable k-medians. *arXiv preprint arXiv:2304.09113*, 2023.

Konstantin Makarychev, Yury Makarychev, Maxim Sviridenko, and Justin Ward. A bi-criteria approximation algorithm for k-means. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2016.

Konstantin Makarychev, Yury Makarychev, and Ilya Razenshteyn. Performance of johnson-lindenstrauss transform for k-means and k-medians clustering. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1027–1038, 2019.

Konstantin Makarychev, Aravind Reddy, and Liren Shan. Improved guarantees for k-means++ and k-means++ parallel. *Advances in Neural Information Processing Systems*, 33, 2020.

Nimrod Megiddo and Kenneth J Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.

Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *IEEE Symposium on Foundations of Computer Science*, pages 165–176, 2006.

Ilias Papanikolaou. Εξηγήσιμη ομαδοποίηση σε ευσταθή στιγμιότυπα. Bachelor's thesis, National Technical University of Athens, 2023.

C Greg Plaxton. Approximation algorithms for hierarchical location problems. *Journal of Computer and System Sciences*, 72(3):425–443, 2006.

J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

J. Ross Quinlan. C4. 5, Programs for machine learning. In *International Conference on Machine Learning*, pages 252–259, 1993.

Sandhya Saisubramanian, Sainyam Galhotra, and Shlomo Zilberstein. Balancing the tradeoff between clustering value and interpretability. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 351–357, 2020.

Dennis Wei. A constant-factor bi-criteria approximation guarantee for k-means++. *Advances in Neural Information Processing Systems*, 29:604–612, 2016.

APPENDIX A

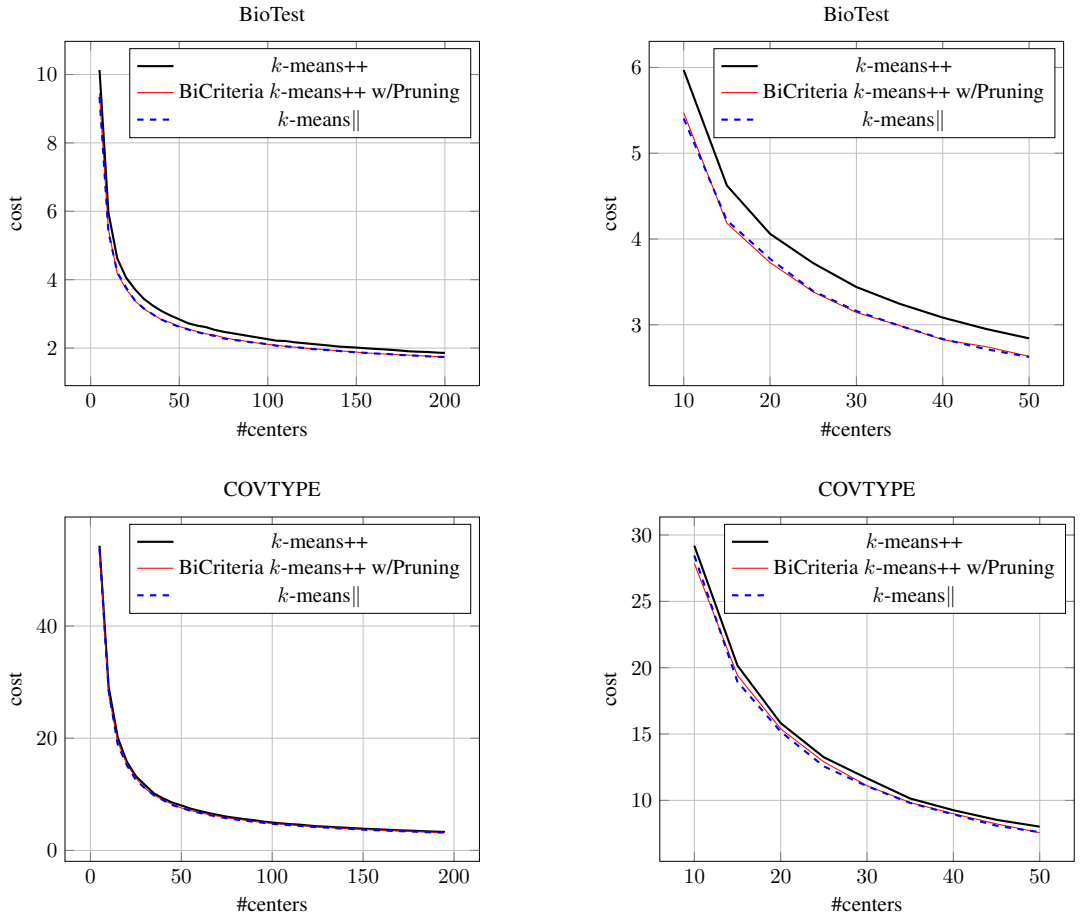
APPENDIX TO CHAPTER 2

A.1 Experiments of k-means++

In this section, we present plots that show that the performance of k-means$\|$ and "k-means++ with oversampling and pruning" algorithms are very similar in practice. Below, we compare the following algorithms on the datasets BioTest from KDD Cup 2004 Elber (2004) and COVTYPE from the UCI ML repository Dua and Graff (2017):

- Regular k-means++. The performance of this algorithm is shown with a solid black line on the plots below.
- k-means$\|$ without pruning. This algorithm samples k centers using k-means$\|$ with $T = 5$ rounds and $\ell = k/T$.
- k-means$\|$. This algorithm first samples $5k$ centers using k-means$\|$ and then subsamples k centers using k-means++. The performance of this algorithm is shown with a dashed blue line on the plots below.
- k-means++ with oversampling and pruning. This algorithm first samples $5k$ centers using k-means++ and then subsamples k centers using k-means++. The performance of this algorithm is shown with a thin red line on the plots below.

For each $k = 5, 10, \cdots, 200$, we ran these algorithms for 50 iterations and took their average. We normalized all costs by dividing them by the cost of k-means++ with $k = 1000$ centers.

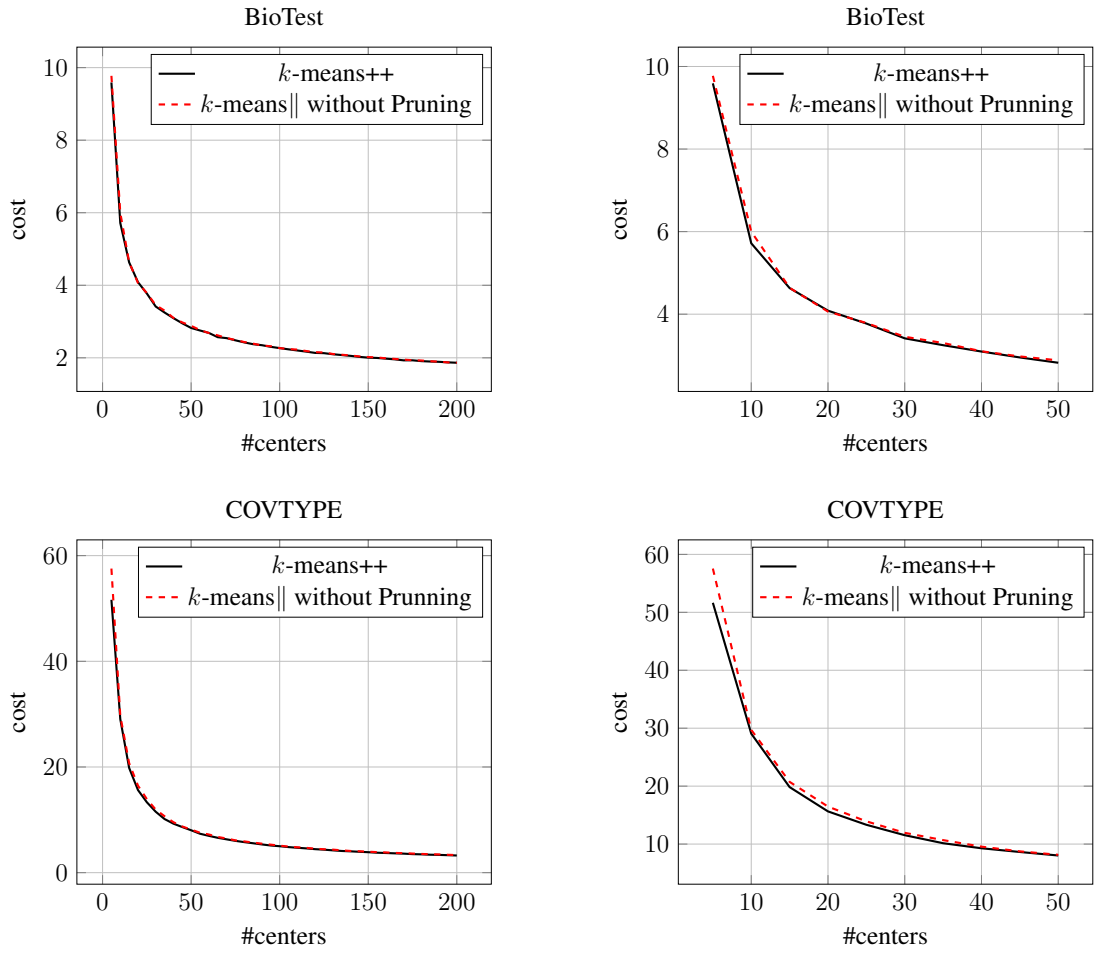


A.2 Lower Bounds for k-means++

A.2.1 Lower Bound on the Cost of Covered Clusters

We show the following lower bound on the expected cost of a covered cluster in k-means++. Therefore, the 5-approximation in Lemma 2.2 is tight.

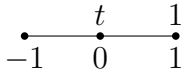
Theorem A.1. *For any $\varepsilon > 0$, there exists an instance of k-means such that for a set $P \in X$ and a set of centers $C \in^d$, if a new center c is sampled from P with probability $\Pr(c = x) =$*



$\mathrm{cost}(x, C)/\mathrm{cost}(P, C)$, *then*

$$\mathbb{E}_c\left[\mathrm{cost}(P, C \cup \{c\})\right] \geq (5 - \varepsilon)\mathrm{OPT}_1(P).$$

Proof. Consider the following one dimensional example, where P contains t points at 0 and one point at 1, and the closest center already chosen in C to P is at -1.



The new center c will be chosen at 0 with probability $\frac{t}{t+4}$, and at 1 with probability $\frac{4}{t+4}$. Then, the expected cost of P is

$$\mathbb{E}_c\left[\text{cost}(P, C \cup \{c\})\right] = 1 \cdot \frac{t}{t+4} + t \cdot \frac{4}{t+4} = \frac{5t}{t+4};$$

and the optimal cost of P is $\text{OPT}_1(P) \leq 1$. Thus, by choosing $t \geq 4(5-\varepsilon)/\varepsilon$, we have

$$\mathbb{E}_c\left[\text{cost}(P, C \cup \{c\})\right] \geq (5-\varepsilon)\text{OPT}_1(P).$$

□

A.2.2 Lower Bound on the Bi-Criteria Approximation

In this section, we show that the bi-criteria approximation bound of $O(\ln \frac{k}{\Delta})$ is tight up to constant factor. Our proof follows the approach by Brunsch and Röglin (2013). We show the following theorem.

Theorem A.2. *For every $k > 1$ and $\Delta \leq k$, there exists an instance X of k-means such that the bi-criteria k-means++ algorithm with $k + \Delta$ centers returns a solution of cost greater than*

$$\frac{1}{8} \log \frac{k}{\Delta} \cdot \text{OPT}_k(X)$$

with probability at least $1 - e^{-\sqrt{k}/2}$.

Remark: This implies that the expected cost of bi-criteria k-means with $k + \Delta$ centers is at least

$$\frac{1 - e^{-\sqrt{k}/2}}{8} \cdot \log \frac{k}{\Delta} \cdot \text{OPT}_k(X).$$

Proof. For every k and $\Delta \geq \sqrt{k}$, we consider the following instance. The first cluster is a scaled version of the standard simplex with $N \gg k$ vertices centered at the origin, which is called the heavy cluster. The length of the edges in this simplex is $1/\sqrt{N-1}$. Each of the remaining $k-1$ clusters contains a single point on $k-1$ axes, which are called light clusters. These clusters are located at distance $\sqrt{\alpha}$ from the center of the heavy cluster and $\sqrt{2\alpha}$ from each other, where $\alpha = \frac{\ln(k/\Delta)}{4\Delta}$.

For the sake of analysis, let us run k-means++ till we cover all clusters. At the first step, the k-means++ algorithm almost certainly selects a center from the heavy cluster since $N \gg k$. Then, at each step, the algorithm can select a center either from one of uncovered light clusters or from the heavy cluster. In the former case, we say that the algorithm hits a light cluster, and in the latter case we say that the algorithm misses a light cluster. Below, we show that with high probability the algorithm makes at least 2Δ misses before it covers all but Δ light clusters.

Lemma A.3. *Let $\Delta \geq \sqrt{k}$. By the time the k-means++ algorithm covers all but Δ light clusters, it makes greater than 2Δ misses with probability at least $1 - e^{-\sqrt{k}/2}$.*

Proof sketch. Let $\varepsilon = 1/\sqrt{N}$. Observe that k-means++ almost certainly covers all clusters in εN steps (since $N \gg k$). So in the rest of this proof sketch, we assume that the number chosen centers is at most εN and, consequently, at least $(1-\varepsilon)N$ points in the heavy cluster are not selected as centers. Hence, the cost of the heavy cluster is at least $1-\varepsilon$.

Consider a step of the algorithm when exactly u light clusters remain uncovered. At this step, the total cost of all light clusters is αu (we assume for simplicity that distance between the light clusters and the closest chosen center in the heavy cluster is the same as the distance to the origin). The cost of the heavy cluster is at least $1-\varepsilon$. The probability that the algorithm chooses a center from the heavy cluster and thus misses a light cluster is at least $(1-\varepsilon)/(1+\alpha u)$.

Define random variables $\{X_u\}$ as follows. Let $X_u = 1$ if the algorithm misses a cluster at least once when the number of uncovered light clusters is u; and let $X_u = 0$, otherwise. Then, $\{X_u\}$ are independent Bernoulli random variables. For each u, we have $\Pr\{X_u = 1\} \geq (1-\varepsilon)/(1+\alpha u)$.

Observe that the total number of misses is lower bounded by $\sum_{u=\Delta}^{k-1} X_u$. Then, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{u=\Delta}^{k-1} X_u\right] &\geq (1-\varepsilon)\sum_{u=\Delta}^{k-1}\frac{1}{1+\alpha u} \geq (1-\varepsilon)\int_{\Delta}^{k}\frac{\mathrm{d}u}{1+\alpha u}\\ &= (1-\varepsilon)\alpha^{-1}\ln\frac{1+\alpha k}{1+\alpha\Delta}\\ &\geq (1-\varepsilon)\alpha^{-1}\ln\frac{k}{\Delta} = 4(1-\varepsilon)\Delta.\end{aligned}$$

Let $\mu = \mathbb{E}\left[\sum_{u=\Delta}^{k-1} X_u\right] \geq 4(1-\varepsilon)\Delta$. By the Chernoff bound for Bernoulli random variables, we have

$$\Pr\left\{\sum_{u=\Delta}^{k} X_u \leq 2\Delta\right\} \leq e^{-\mu}\left(\frac{e\mu}{2\Delta}\right)^{2\Delta}.$$

Since $f(x) = e^{-x}(\frac{ex}{2\Delta})^{2\Delta}$ is a monotone decreasing function for $x \geq 2\Delta$, we have

$$\Pr\left\{\sum_{u=\Delta}^{k} X_u \leq 2\Delta\right\} \leq e^{-(2-4\varepsilon)\Delta}\cdot 2^{2\Delta} \leq e^{-\Delta/2}.$$

Hence, with probability as least $1 - e^{-\sqrt{k}/2}$, the number of misses is greater than 2Δ. □

For every k and $\Delta \geq \sqrt{k}$, consider the instance we constructed. By Lemma A.3, the algorithm chooses more than $k+\Delta$ centers to cover all but Δ light clusters with probability at least $1-e^{-\sqrt{k}/2}$. Thus, at the time when the algorithm chose $k+\Delta$ centers, the number of uncovered light clusters was greater than Δ. Hence, in the clustering with $k+\Delta$ centers sampled by k-means++, the total cost is at least $\frac{1}{4}\ln(k/\Delta)$, while the cost of the optimal solution with k clusters is 1. For every k and $\Delta < \sqrt{k}$, the total cost is at least $\frac{1}{4}\ln(k/\Delta')$ with $\Delta' = \sqrt{k}$ extra centers, which concludes the

proof. □

APPENDIX B

APPENDIX TO CHAPTER 3 AND CHAPTER 4

B.1 Lower Bound for Threshold Tree

B.1.1 Lower Bound for k-means

In this section, we show a lower bound on the price of explainability for k-means.

Theorem B.1. *For any k, there exists an instance X with k clusters such that the cost of explainable k-means clustering for every tree T is at least*

$$\mathrm{cost}_{\ell_2^2}(X, T) \geq \Omega\left(\frac{k}{\log k}\right) \mathrm{OPT}_{\ell_2^2}(X).$$

To prove this lower bound, we construct an instance as follows. We uniformly sample k centers $C = \{c^1, c^2, \cdots, c^k\}$ from the d-dimensional unit cube $[0, 1]^d$ where the dimension $d = 300 \ln k$. For each center c^i, we add two points $c^i \pm (\varepsilon, \varepsilon, \cdots, \varepsilon)$ with $\varepsilon = 300 \ln k/k$. We also add many points at each center such that the optimal centers for any threshold tree remain almost the same. Specially, we can add k^2 points co-located with each center c^i. Then, if one center c^i is shifted by a distance of ε in the threshold tree clustering, the cost of the co-located points at c^i is at least $k^2\varepsilon^2$. Since the optimal regular cost for this instance is $kd\varepsilon^2$, the total cost of the threshold tree is lower bounded by $\Omega(k/\log k)\mathrm{OPT}_{\ell_2^2}(X)$. Consequently, we consider the threshold tree with optimal centers shifted by at most ε.

First, we show that any two centers defined above are far apart with high probability.

Lemma B.2. *With probability at least $1-1/k^2$ the following holds: The squared distance between every two distinct centers c and c' in C is at least $d/12$.*

Proof. Consider any fixed two centers $c, c' \in C$. Since c, c' are uniformly sampled from $[0,1]^d$, each coordinate of c, c' is sampled from $[0,1]$; and centers c, c' are sampled independently. Thus, we have

$$\mathbb{E}_{c,c'}[\|c-c'\|^2] = \sum_{i=1}^{d} \mathbb{E}_{c_i,c_i'}[(c_i-c_i')^2] = \frac{d}{6}.$$

We use a random variable X_i to denote $(c_i - c_i')^2$ for each coordinate $i \in \{1,\ldots,d\}$. Since random variables $\{X_i\}_{i=1}^d$ are independent, by Hoeffding's inequality, we have

$$\Pr\left\{\sum_{i=1}^{d} X_i - \mathbb{E}\Big[\sum_{i=1}^{d} X_i\Big] \le -\sqrt{2d\ln k}\right\} \le e^{-4\ln k} = \frac{1}{k^4},$$

where we used that $d = 300\ln k$. This implies that the squared distance between c and c' is less than $d/12$ with probability at most $1/k^4$. Using the union bound over all pairs of centers in C, we conclude that the squared distance between all pairs in C is at least $d/12$ with probability at least $1-1/k^2$. □

If any two centers are far apart, then a point x separated from its original center will incur a large penalty. Thus, we can get a lower bound if there exists an instance which satisfies: (1) any two centers are separated by a large distance; (2) every threshold tree separates a relatively large portion of points from their original centers. In particular, we prove that with probability $1-o(1)$, every threshold cut separates a relatively large portion of points from their original centers in the random instance we constructed.

Lemma B.3. *With probability at least $1-1/k^2$, the following holds: every threshold cut (i,θ) with $i \in \{1,2,\cdots,d\}$ and $\theta \in [0,1)$ separates at least $\varepsilon k/4$ points from their original centers.*

Proof. Consider a fixed coordinate $i \in \{1, \ldots, d\}$. We project each center and its rectangular neighborhood onto this coordinate. For each center $c^j \in C$, we define an interval I_i^j as the intersection of $[0, 1]$ and the ε-neighborhood of its projection c_i^j, i.e. $I_i^j = (c_i^j - \varepsilon, c_i^j + \varepsilon) \cap [0, 1]$. Each interval I_i^j has length at least ε. If we pick a threshold cut inside any interval I_i^j, then we separate at least one points from center c^j. In this case, the interval I_i^j is called covered by this threshold cut. Then, we give the lower bound on the minimum number of intervals covered by a threshold cut.

For a fixed set of centers C, we consider at most $2k$ special positions for the threshold cut at coordinate i as follows. Let E_i be the set containing two end points of intervals I_i^j for all centers c^j. For any threshold cut at coordinate i, the closest position in set E_i covers exactly the same set of intervals as this threshold cut. Thus, we only need to consider threshold cuts at positions in E_i.

For centers chosen uniformly from $[0, 1]^d$, the set E_i contains $2k$ random variables. Suppose we pick a threshold cut at a position θ in E_i related to interval I_i^j. Conditioned on the position θ, the other $k - 1$ centers c^j for $j \neq j^*$ are uniformly distributed in $[0, 1]^d$ since all centers are chosen independently. For $j \in \{1, 2, \cdots, k\} \setminus \{j^*\}$, let Y_i^j be the indicator random variable that the interval I_i^j contains this position θ. For each variable Y_i^j, we have $\varepsilon \leq \Pr\{Y_i^j = 1\} \leq 2\varepsilon$. Since random variables Y_i^j are independent, by the Chernoff bound for Bernoulli random variables, we have

$$\Pr\left\{\sum_j Y_i^j - \mathbb{E}\left[\sum_j Y_i^j\right] \leq -\sqrt{18\varepsilon k \ln k} \mid \theta\right\} \leq e^{-4\ln k} = \frac{1}{k^4}.$$

Thus, we have the number of intervals containing this position θ is at least $\varepsilon k/4$ with probability at least $1 - 1/k^4$.

Since we have $2k$ positions E_i for each coordinate $i \in \{1, 2, \cdots, d\}$, there are total $2dk$ positions for threshold cuts. Using the union bound over all positions, we have the minimum number of intervals covered by a threshold cut is at least $\varepsilon k/4$ with probability at least $1 - 1/k^2$. Since the

threshold cut separates one point from its original center for each covered interval, we have every threshold cut separates at least $\varepsilon k/4$ points from their original centers in this case. □

Proof of Theorem B.1. By Lemma B.2, we can only consider the instance where any two centers are separated with the squared distance at least $d/12$. Note that the optimal centers for any threshold tree remain almost the same as centers C. Thus, we analyze the k-means cost given by any threshold tree with respect to center C. If a point in X is separated from its original center, this point will finally be assigned to another center in C. By the triangle inequality, the k-means cost of this point is at least $d/20$. By Lemma B.3, there exists an instance such that any threshold cut separates at least $\varepsilon k/4$ points from their original centers. Thus, there exists an instance X such that any threshold tree T has the k-means cost at least

$$\mathrm{cost}_{\ell_2^2}(X, T) \geq \frac{\varepsilon k}{4} \cdot \frac{d}{20} = \frac{\varepsilon k d}{80}.$$

Note that the optimal regular k-means cost for this instance X is

$$\mathrm{OPT}_{\ell_2^2}(X) = 2k \cdot \varepsilon^2 d.$$

Therefore, the k-means cost for this instance X given by any threshold tree T is at least

$$\mathrm{cost}_{\ell_2^2}(X, T) \geq \frac{1}{160\varepsilon} \cdot \mathrm{OPT}_{\ell_2^2}(X) = \Omega\left(\frac{k}{\log k}\right) \cdot \mathrm{OPT}_{\ell_2^2}(X).$$

□

B.1.2 Lower Bound for k-medians in ℓ_2

In this section, we show a lower bound on the price of explainability for k-medians in ℓ_2.

Theorem B.4. *For every $k \geq 1$, there exists an instance X with k clusters such that the k-medians with ℓ_2 objective cost of every threshold tree T is at least*

$$\mathrm{cost}_{\ell_2}(X, T) \geq \Omega(\log k)\mathrm{OPT}_{\ell_2}(X).$$

To prove this lower bound, we use the construction similar to that used in Theorem B.1. We discretize the d-dimensional unit cube $[0, 1]^d$ into grid with length $\varepsilon = 1/\lceil \ln k \rceil$, where the dimension $d = 300 \ln k$. We uniformly sample k centers $C = \{c^1, c^2, \cdots, c^k\}$ from the above grid $\{0, \varepsilon, 2\varepsilon, \cdots, 1\}^d$. For each center c^i, we add 2 points $c^i \pm (\varepsilon, \varepsilon, \cdots, \varepsilon)$ to this center. Similar to Theorem B.1, we also add many points at each center such that the optimal centers for any threshold tree remain almost the same.

Similar to Lemma B.2, we show that any two centers defined above are far apart with high probability.

Lemma B.5. *With probability at least $1 - 1/k^2$ the following holds: The distance between every two distinct centers c and c' in C is at least $\sqrt{d}/4$.*

Proof. To sample a center from the grid uniformly, we can first sample a candidate center uniformly from the cube $[-\varepsilon/2, 1 + \varepsilon/2]^d$ and then move it to the closest grid point. Note that the ℓ_2-distance from every point in this cube to its closest grid point is at most $\varepsilon\sqrt{d} = o(1)$. By Lemma B.2, the ℓ_2 distance between every pairs of candidate centers is at least $\sqrt{d/12}$ with probability at least $1 - 1/k^2$. Thus, the distance between every two distinct centers is at least $\sqrt{d}/4$ with probability at least $1 - 1/k^2$. □

For every node in the threshold tree, we can specify it by threshold cuts in the path from the root to this node. Thus, we define a path π as an ordered set of tuples $(i_j, \theta_j, \sigma_j)$, where (i_j, θ_j) denotes the j-th threshold cut in this path and $\sigma_j \in \{\pm 1\}$ denotes the direction with respect to

this cut. We use $u(\pi)$ be the node specified by the path π. We define a center is damaged if one of its two points are separated by this cut, otherwise a center is undamaged. Let F_u be the set of undamaged centers in node u.

Lemma B.6. *With probability at least $1 - 1/k$, the following holds: For every path π with length less than $\log_2 k/4$, we have (a) the node $u(\pi)$ contains at most $\sqrt{k}$ undamaged centers; or (b) every cut in node $u(\pi)$ damages at least $\varepsilon|F_{u(\pi)}|/2$ centers in $F_{u(\pi)}$.*

Proof. Consider any fixed path π with length less than $\log_2 k/4$. We upper bound the probability that both events (a) and (b) do not happen conditioned on $F_{u(\pi)}$. If $|F_{u(\pi)}| \leq \sqrt{k}$, then the event (a) happens. For the case $F_{u(\pi)}$ contains more than $\sqrt{k}$ centers, we pick an arbitrary threshold cut (i, θ) in the node $u(\pi)$. For every center c in $F_{u(\pi)}$, the probability we damage this center c is at least ε. Let X_j be the indicator random variable that the j-th center in $F_{u(\pi)}$ is damaged by the threshold cut (i, θ). Then, we have the expected number of centers in $F_{u(\pi)}$ damaged by this cut (i, θ) is

$$\mathbb{E}\left[\sum_j X_j\right] \geq \varepsilon|F_{u(\pi)}|.$$

Let $\mu = \mathbb{E}[\sum_j X_j]$. By the Chernoff bound for Bernoulli random variables, we have

$$\Pr\left\{\sum_j X_j \leq \varepsilon|F_{u(\pi)}|/2\right\} \leq \Pr\left\{\sum_j X_j \leq \mu/2\right\} \leq e^{-\mu/8} \leq e^{-\varepsilon\sqrt{k}/8}.$$

Using the union bound over all threshold cuts in $u(\pi)$, the failure probability that both event (a) and (b) do not happen is at most $e^{-\varepsilon\sqrt{k}/16}$. The number of paths with length less than $\log_2 k/4$ is at most $m(2d/\varepsilon)^m \leq e^{-\log^2 k}$. Thus, by the union bound over all paths with length less than $\log_2 k/4$, we get the conclusion. □

Proof of Theorem B.4. By Lemma B.5 and Lemma B.6, we can find an instance X such that both

two properties hold. We first show that the threshold tree must separate all centers. Suppose there is a leaf contains more than one center. Since the distance between every two centers is at least $\sqrt{d}/4$ and there are many points at each center, the cost for this leaf can be arbitrary large. To separate all centers, the depth of the threshold tree is at least $\lceil \log_2 k \rceil$.

We now lower bound the cost for every threshold tree that separates all centers. Consider any threshold tree T that separates all centers. We consider the following two cases. If the number of damaged centers at level $\lfloor \log_2 k \rfloor/4$ of threshold tree T is more than $k/2$, then the cost given by T is at least

$$\mathrm{cost}_{\ell_2}(X, T) \geq \frac{k}{2} \cdot \frac{\sqrt{d}}{8} = \frac{k\sqrt{d}}{16}.$$

If the number of damaged centers at level $\lfloor \log_2 k \rfloor/4$ of threshold tree T is less than $k/2$, then the number of undamaged centers at every level $i = 1, 2, \ldots, \lfloor \log_2 k \rfloor/4$ is at least $k/2$. We call a node u a small node if it contains at most $\sqrt{k}$ undamaged centers, otherwise we call it a large node. Then, we lower bound the number of damaged centers generated at any fixed level $i \in \{1, 2, \cdots, \lfloor \log_2 k \rfloor/4\}$. Since the number of nodes at level i is at most $k^{1/4}$, the number of undamaged centers in small nodes at level i is at most $k^{3/4}$. Thus, the number of undamaged centers in large nodes at level i is at least $k/4$. By Lemma B.6, the number of damaged centers generated at level i is at least $\varepsilon k/8$. Therefore, the cost given by this threshold tree T is at least

$$\mathrm{cost}_{\ell_2}(X, T) \geq \frac{\lfloor \log_2 k \rfloor}{4} \frac{\varepsilon k}{8} \frac{\sqrt{d}}{8} = \Omega(k\sqrt{d}\varepsilon \log k).$$

Note that the optimal cost for this instance is at most $k\varepsilon\sqrt{d}$ and $\varepsilon = 1/\lceil \log k \rceil$. Combining the two cases above, we have the cost given by threshold tree T is at least

$$\mathrm{cost}_{\ell_2}(X, T) = \Omega(k\sqrt{d}\varepsilon \log k) = \Omega(\log k)\mathrm{OPT}_{\ell_2}(X).$$

□

B.1.3 Lower Bound on the Bi-criteria Approximation for k-means

In this section, we prove Theorem 4.2. We show a lower bound on the price of explainability for k-means in the bi-criteria setting. Our proof follows the general approach by Makarychev and Shan (2021).

Theorem 4.2. *For every $k > 500$ and $\ln^3 k/\sqrt{k} < \delta < 1/100$, there exists an instance X with k clusters such that the k-means cost for every threshold tree $\mathcal{T}$ with $(1+\delta)k$ leaves is at least*

$$\operatorname{cost}(X, \mathcal{T}) \geq \Omega\left(\frac{\log^2 k}{\delta}\right) \operatorname{OPT}_k(X).$$

Proof of Theorem 4.2. We construct a hard instance for explainable clustering as follows. Let $d = 300\lceil \ln k \rceil$. Consider the grid $\{0, \varepsilon, 2\varepsilon, \dots, 1\}^d$ with step size $\varepsilon = 50\delta/\lceil \ln k \rceil$ in the d-dimensional unit cube $[0,1]^d$. We uniformly sample k centers $C = \{c^1, c^2, \dots, c^k\}$ from the nodes of the grid. Then, we create a data set X. For every center c^i in C, data set X contains many (namely, $k^2\lceil \ln^3 k \rceil$) points co-located with c^i and two special points $c^i \pm (\varepsilon, \varepsilon, \dots, \varepsilon)$. Hence, the total number of points in X is $k^3\lceil \ln^3 k \rceil + 2k$. Note that all centers and all points in X lie in the nodes of the grid.

The cost of the k-means clustering with centers $C = \{c^1, c^2, \dots, c^k\}$ equals $2kd\varepsilon^2$, since the distance from the special points $c^i \pm (\varepsilon, \varepsilon, \dots, \varepsilon)$ to c^i is $\varepsilon\sqrt{d}$. Hence, the cost of the optimal k-means clustering is at most $2kd\varepsilon^2$. We now show that there exists an instance such that the cost of every *explainable* k-means clustering with $(1+\delta)k$ centers is at least $2kd\varepsilon^2 \cdot \Omega(1/\delta \log^2 k)$. In this instance, every explainable k-means clustering with $(1+\delta)k$ centers separates at least $\delta k = \Omega(\varepsilon k \ln k)$ special points $c^i \pm (\varepsilon, \varepsilon, \dots, \varepsilon)$ from c^i. The cost of each special point separated

from its original center is at least $\Omega(d)$. Thus, the total cost of every explainable k-means clustering is at least $\Omega(d\varepsilon k \ln k) = 2kd\varepsilon^2 \cdot \Omega(1/\delta \log^2 k)$. First, we prove that with high probability every two centers in C are far apart.

Lemma B.7. *With probability at least $1 - 1/k^2$ the following statement holds: The distance between every two distinct centers c' and c'' in C is at least $\sqrt{d}/5$.*

Proof. We can select a random center in the grid $\{0, \varepsilon, 2\varepsilon, \ldots, 1\}^d$ using the following procedure: First, pick a candidate center uniformly from the cube $[-\varepsilon/2, 1 + \varepsilon/2]^d$ and then move the chosen point to the closest grid point. Note that the ℓ_2-distance from every point in this cube to the closest grid point is at most $\sqrt{d}\,\varepsilon/2 \le \sqrt{d}/36$ since $\varepsilon \le 1/18$.

Consider two distinct centers $c', c'' \in C$. Let c^* and c^{**} be the candidate centers corresponding to c' and c''. If $\|c^* - c^{**}\|_2 \ge \sqrt{d/12}$, then by the triangle inequality, we have

$$\|c' - c''\|_2 \ge \|c^* - c^{**}\|_2 - \|c' - c^*\|_2 - \|c'' - c^{**}\|_2 \ge \frac{\sqrt{d}}{\sqrt{12}} - \frac{\sqrt{d}}{18} \ge \frac{\sqrt{d}}{5}.$$

Thus, we need to show that with probability at least $1 - 1/k^2$, the ℓ_2-distance between every two candidate centers uniformly sampled from the cube $[-\varepsilon/2, 1 + \varepsilon/2]^d$ is at least $\sqrt{d/12}$. Consider two candidate centers c^*, c^{**}. Since c^*, c^{**} are chosen uniformly from $[-\varepsilon/2, 1 + \varepsilon/2]^d$, each coordinate of c^*, c^{**} is drawn from $[-\varepsilon/2, 1 + \varepsilon/2]$. Hence, we have

$$\mathbb{E}_{c^*, c^{**}}[\|c^* - c^{**}\|_2^2] = \sum_{i=1}^{d} \mathbb{E}_{c_i^*, c_i^{**}}[(c_i^* - c_i^{**})^2] = d \cdot \frac{(1+\varepsilon)^2}{6}.$$

Let $X_i = (c_i^* - c_i^{**})^2/(1+\varepsilon)^2$ for $i \in \{1, \ldots, d\}$. Random variables $\{X_i\}_{i=1}^d$ are independent and

each X_i lies in $[0, 1]$. Thus, by Hoeffding's inequality, we have

$$\Pr\left\{\sum_{i=1}^{d} X_i - \mathbb{E}\left[\sum_{i=1}^{d} X_i\right] \leq -\sqrt{2d \ln k}\right\} \leq e^{-4 \ln k} = \frac{1}{k^4}.$$

Since $d = 300\lceil \ln k \rceil$, the squared distance between c^* and c^{**} is less than $d/12$ with probability at most $1/k^4$. Using the union bound over all pairs of candidate centers, we conclude that the squared distance between every two candidate centers is at least $d/12$ with probability at least $1 - 1/k^2$. $\square$

All data points in X are in the grid $\{-\varepsilon, 0, \varepsilon, 2\varepsilon, \ldots, 1, 1 + \varepsilon\}^d$. Every internal node u in the threshold tree should contain a threshold cut that separates at least two data points in that node u. Otherwise, we can ignore this threshold cut since one side of this cut contains no data points. If two threshold cuts have the same coordinate and thresholds within the same grid interval $(j\varepsilon, j\varepsilon + \varepsilon)$, then these two threshold cuts create the same partition of data points contained in the internal node. Since there are at most $1/\varepsilon + 2$ different grid intervals for each coordinate, the number of distinct threshold cuts for each internal node is at most $d(1/\varepsilon + 2) \leq 2d/\varepsilon$. Every node in the threshold tree corresponds to a cell in $\mathbb{R}^d$. This cell is determined by the threshold cuts on the path from the root to that node. Let π be an ordered set of tuples (i_j, ξ_j, λ_j), where (i_j, ξ_j) is the j-th threshold cut on the path from the root to the node, and $\lambda_j \in \{\pm 1\}$ specifies one of the sides of the cut. Then, every ordered set π corresponds to a path in the threshold tree starting in the root.

Let $u(\pi)$ be the intersection of the cuts in π. We say that a center c^i in $u(\pi)$ is damaged if one of the special points $c^i \pm (\varepsilon, \ldots, \varepsilon)$ is separated from c^i by one of the threshold cuts in π. In other words, c^i is damaged if $c^i \in u(\pi)$, but $c^i - (\varepsilon, \ldots, \varepsilon) \notin u(\pi)$ or $c^i + (\varepsilon, \ldots, \varepsilon) \notin u(\pi)$. Otherwise, we say that c^i is not damaged. Similarly, we say that a node of the grid $x \in u(\pi)$ is not damaged if $x \pm (\varepsilon, \ldots, \varepsilon) \in u(\pi)$. Let $F_{u(\pi)}$ be the set of all centers that are not damaged in node $u(\pi)$. We show that with high probability, if a node $u(\pi)$ contains more than $\sqrt{k}$ centers, every threshold cut

that splits node $u(\pi)$ damages at least $\varepsilon|F_{u(\pi)}|/2$ centers in $F_{u(\pi)}$.

Lemma B.8. *With probability at least $1 - 1/k$, the following holds: For every path (ordered set of cuts) π of length at most $\log_2 k/4$, we have (a) $|F_{u(\pi)}| \leq \sqrt{k}$; or (b) every threshold cut that separates at least two data points in $u(\pi)$ damages at least $\varepsilon|F_{u(\pi)}|/2$ centers in $F_{u(\pi)}$.*

Proof. Consider a fixed ordered set of cuts π of size at most $\log_2 k/4$. We upper bound the probability that both events (a) and (b) do not occur for this fixed path π on the random instance X. If $|F_{u(\pi)}| \leq \sqrt{k}$, then the event (a) happens. So, we assume that $F_{u(\pi)}$ contains more than $\sqrt{k}$ centers. We then bound the probability that event (b) happens conditioned on the size of $F_{u(\pi)}$. Observe that all centers in $F_{u(\pi)}$ are distributed uniformly and independently among the grid nodes in $u(\pi)$ that are not damaged by the cuts in π conditioned on $|F_{u(\pi)}|$. Pick an arbitrary threshold cut (i, ξ) in $u(\pi)$ that separates at least two nodes of the grid in $u(\pi)$. For every center c in $F_{u(\pi)}$, the probability that the threshold cut (i, ξ) damages this center c is at least ε. Let X_j be the indicator random variable that the j-th center in $F_{u(\pi)}$ is damaged by (i, ξ). The expected number of centers in $F_{u(\pi)}$ damaged by cut (i, ξ) conditioned on $|F_{u(\pi)}| = l$ equals

$$\mathbb{E}\left[\sum_{j=1}^{l} X_j \;\middle|\; |F_{u(\pi)}| = l\right] \geq \varepsilon l.$$

Let $\mu = \mathbb{E}[\sum_j X_j \mid |F_{u(\pi)}| = l]$. By the Chernoff bound for Bernoulli random variables, we have

$$\Pr\left\{\sum_{j=1}^{l} X_j \leq \varepsilon|F_{u(\pi)}|/2 \;\middle|\; |F_{u(\pi)}| = l\right\} \leq \Pr\left\{\sum_{j=1}^{l} X_j \leq \mu/2 \;\middle|\; |F_{u(\pi)}| = l\right\} \leq e^{-\mu/8} \leq e^{-\varepsilon\sqrt{k}/8}.$$

Combining all conditional probabilities for $|F_{u(\pi)}| > \sqrt{k}$, the probability that the event (b) doesn't happen is at most $e^{-\varepsilon\sqrt{k}/8}$. Since all data points are in the grid $\{-\varepsilon, 0, \varepsilon, 2\varepsilon, \ldots, 1, 1+\varepsilon\}^d$, there are at most $2d/\varepsilon$ different threshold cuts that separates at least two data points in node $u(\pi)$. By the

union bound, the probability that both events (a) and (b) do not happen is at most $e^{-\varepsilon\sqrt{k}/8} \cdot 2d/\varepsilon \leq e^{-2\ln^2 k}$. Since there are at most $4d/\varepsilon$ different choices for each tuple (i_j, ξ_j, λ_j) in π, the number of paths with length less than $m = \log_2 k/4$ is at most $m(4d/\varepsilon)^m \leq e^{\ln^2 k}$. Thus, by the union bound over all paths with length less than $\log_2 k/4$, we get that (a) or (b) holds with probability at least

$$1 - m(4d/\varepsilon)^m \cdot e^{-\varepsilon\sqrt{k}/8} \cdot 2d/\varepsilon \geq 1 - e^{\ln^2 k} \cdot e^{-2\ln^2 k} \geq 1 - \frac{1}{k}.$$

since $d/\varepsilon \leq 15000\sqrt{k}\ln^3 k$ for $d = 300\lceil \ln k \rceil$ and $\varepsilon = 50\delta/\lceil \ln k \rceil \geq 50\sqrt{k}\ln^2 k$. □

By Lemma B.7 and Lemma B.8, we can find an instance X such that the following conditions hold:

- The distance between every two distinct centers c' and c'' in C is at least $\sqrt{d}/5$.
- For every path (ordered set of cuts) π of length at most $\log_2 k/4$, we have (a) $|F_{u(\pi)}| \leq \sqrt{k}$; or (b) every threshold cut that separates at least two data points in $u(\pi)$ damages at least $\varepsilon|F_{u(\pi)}|/2$ centers in $F_{u(\pi)}$.

We first show that the threshold tree must separate all centers. Suppose there is a leaf contains more than one center. Since the distance between every two centers is at least $\sqrt{d}/5$, there exists at least one center in this leaf with distance greater than $\sqrt{d}/10$ to the optimal center of this leaf. Since we add $k^2\lceil \ln^3 k \rceil$ points co-located with each center, the cost for the leaf that contains more than one center is greater than $k^2\lceil \ln^3 k \rceil \cdot d/100 = 2kd\varepsilon^2 \cdot \Omega(1/\delta \log^2 k)$. Thus, the lower bound holds for any threshold tree that does not separate all centers. To separate all centers, the depth of the threshold tree must be at least $\lceil \log_2 k \rceil$. We show the following lower bound on the number of damaged centers for every threshold tree that separates all centers.

Lemma B.9. *Consider any instance X with k centers satisfies two conditions in Lemma B.7 and Lemma B.8. For every threshold tree that separates all centers in C, there are at least $2\delta k$ damaged centers.*

Proof. Consider any threshold tree $\mathcal{T}$ that separates all centers. We consider the following two cases. If the number of damaged centers at level $\lfloor \log_2 k \rfloor / 4$ of threshold tree $\mathcal{T}$ is more than $k/2$, then the total number of damaged centers generated by this threshold tree is more than $2\delta k$.

If the number of damaged centers at level $\lfloor \log_2 k \rfloor / 4$ of threshold tree $\mathcal{T}$ is less than $k/2$, then the number of centers that are not damaged at each level $i = 1, 2, \ldots, \lfloor \log_2 k \rfloor / 4$ is at least $k/2$. We call a node u a small node if it contains at most $\sqrt{k}$ centers which are not damaged, otherwise we call it a large node. We now lower bound the number of centers damaged at a fixed level $i \in \{1, 2, \cdots, \lfloor \log_2 k \rfloor / 4\}$. For every level $i \in \{1, 2, \cdots, \lfloor \log_2 k \rfloor / 4\}$, the number of nodes at level i is at most $k^{1/4}$. Since each small node contains at most $\sqrt{k}$ centers that are not damaged, the total number of centers that are not damaged in small nodes at level i is at most $k^{3/4}$. Since the total number of centers that are not damaged at level i is at least $k/2$, the number of centers that are not damaged in large nodes at level i is at least $k/4$. By Lemma B.8, the number of damaged centers generated at level i is at least $\varepsilon k/8$. Therefore, the total number of damaged centers generated by this threshold tree $\mathcal{T}$ is at least

$$\frac{\lfloor \log_2 k \rfloor}{4} \cdot \frac{\varepsilon k}{8} \geq \frac{50 \lfloor \log_2 k \rfloor \delta k}{32 \ln k} \geq 2\delta k,$$

which completes the proof. □

We now lower bound the cost for every threshold tree with $(1+\delta)k$ leaves that separates all centers. Consider any threshold tree $\mathcal{T}$ with $(1+\delta)k$ leaves that separates all centers in C. By Lemma B.9, we have more than $2\delta k$ data points separated from their original centers by $\mathcal{T}$. For

each point x separated from its original center c, one and only one of the following may occur: (1) the data point x is assigned to a leaf containing a center $c' \neq c$; (2) the data point x is assigned to a leaf containing no center. Among these $2\delta k$ data points, we show that there are at least δk data points that have distances to their new centers greater than $\sqrt{d}/20$.

For each leaf containing a center c', the optimal center for this leaf is shifted from c' by at most $\varepsilon\sqrt{d}$. Otherwise, the cost of this leaf is at least $k^2\lceil \ln^3 k \rceil \cdot \varepsilon^2 d = 2kd\varepsilon^2 \cdot \Omega(1/\delta \log^2 k)$ since there are $k^2\lceil \ln^3 k \rceil$ data points co-located at each center. Suppose a point x separated from its original center c is assigned to a leaf containing a center $c' \neq c$. By Lemma B.7 and the triangle inequality, the distance from the point x to the optimal center for this leaf is at least $\sqrt{d}/10$.

For each leaf containing no center, it may contain several points from distinct clusters. Among these points, there is at most one point within $\sqrt{d}/20$ distance of the optimal center for this leaf. Suppose two points x' and x'' from distinct clusters are within $\sqrt{d}/20$ distance of the optimal center for this leaf. Then, the distance between x' and x'' is at most $\sqrt{d}/10$. Let c' and c'' be the original centers for points x' and x'' respectively. The distance between c' and c'' is at most $\sqrt{d}/10 + 2\varepsilon\sqrt{d} \leq \sqrt{d}/5$, which contradicts the distance between every two centers is at least $\sqrt{d}/5$.

Since the threshold tree $\mathcal{T}$ has $(1+\delta)k$ leaves, there are δk leaves that do not contain a center. Thus, among points separated from their original centers, there are at most δk points with distance less than $\sqrt{d}/20$ to their new centers. Since there are more than $2\delta k$ points separated from their original centers, we have at least δk points with cost greater than $d/400$. Therefore, the cost given by this threshold tree $\mathcal{T}$ is at least

$$\mathrm{cost}(X, \mathcal{T}) \geq \delta k \cdot \frac{d}{400} = \Omega(\delta d k).$$

Recall that the optimal k-means cost for this instance is at most $2k\varepsilon^2 d$ and $\varepsilon = 50\delta/\lceil \ln k \rceil$. Thus, the cost given of this explainable clustering is at least

$$\text{cost}(X, \mathcal{T}) = \Omega(\delta dk) \geq \Omega\left(\frac{\log^2 k}{\delta}\right) \text{OPT}_k(X).$$

□

B.1.4 Lower Bound for the ExKMC Algorithm

In this section, we show the lower bound for the ExKMC algorithm. The ExKMC algorithm is an expanding explainable k-means algorithm proposed by Frost et al. (2020). Given a parameter $k' > k$ as the number of leaves, the ExKMC outputs a threshold tree T with k' leaves. We consider the ExKMC algorithm that starts from the base tree given by the IMM algorithm in Dasgupta et al. (2020). The IMM algorithm iteratively chooses the threshold cut that minimizes the number of mistakes, where a mistake means a point is separated from its original center. For any threshold tree with more than k leaves, the ExKMC algorithm considers the surrogate cost, which is the cost by assigning each leaf to its best center in C. Then, the ExKMC algorithm iteratively chooses the threshold cut that minimizes the surrogate cost. Our proof is inspired by the constructions in Esfandiari et al. (2022), Laber and Murtinho (2021) and Charikar and Hu (2022).

Theorem B.10. *For every $k > 10$, and $\delta \in (0, 1/4)$, there exists an instance X with k clusters such that the k-means cost for the threshold tree $\mathcal{T}$ returned by the ExKMC algorithm with an IMM base tree and $k' = (1+\delta)k$ leaves is at least*

$$\text{cost}(X, \mathcal{T}) \geq \Omega\left((1 - 4\delta) \cdot \frac{k^2}{\log k}\right) \text{OPT}_k(X).$$

Remark: This provides a $\tilde{\Omega}(k^2)$ lower bound for the ExKMC algorithm when $\delta \in (0, 1)$ is a

constant and $k \to \infty$.

Proof. We first construct k centers for the instance. Without loss of generality, we assume $k = 2\tilde{k} + 1$ is an odd number. Let $p = 3\log_2 k$ and $d = \tilde{k} + p - 1$. Then, we choose k centers $C = \{c^1, c^2, \cdots, c^k\}$ in the d-dimensional space $\mathbb{R}^d$. Let the first $\tilde{k} - 1$ coordinates of c^1 be all zeros $(0, 0, \cdots, 0)$. For each $i \in \{2, \cdots, \tilde{k} + 1\}$, let the first $\tilde{k} - 1$ coordinates of center c^i be the same as those of e_{i-1} the identity vector on the $(i-1)$-th coordinate. For every coordinate $j \in \{\tilde{k}, \tilde{k} + 1, \cdots, \tilde{k} + p - 1\}$, we pick a random permutation σ of $\{0, 1, \cdots, \tilde{k}\}$ and assign the j-th coordinate of centers $c^1, c^2, \ldots, c^{\tilde{k}+1}$ be this random permutation, i.e. $c^i_j = \sigma(j)$. For each $i \in \{\tilde{k} + 2, \cdots, 2\tilde{k} + 1\}$, the first $\tilde{k} - 1$ coordinates of c^i are all zero, and the rest p coordinates of center c^i are identical to those of the center $c^{i-\tilde{k}}$.

We now construct the instance X as follows. For the center c^1 and every coordinate $j \in \{1, 2, \cdots, \tilde{k} - 1\}$, we add one data point at e_j. For every center c^i, and every coordinate $j \in \{\tilde{k}, \tilde{k} + 1, \cdots, \tilde{k} + p - 1\}$, we add two data points at $c^i + e_j$ and two data points at $c^i - e_j$. For every center c^i, we also add many data points co-located with c^i.

For this instance X, the cost of the k-means clustering with centers $C = \{c^1, c^2, \cdots, c^k\}$ equals $(\tilde{k} - 1) + 4p\tilde{k}$. Thus, the optimal k-means cost of X is at most $(\tilde{k} - 1) + 4p\tilde{k} = O(\tilde{k} \log k)$. Let $\mathcal{T}$ be the threshold tree returned by the ExKMC algorithm with the IMM base tree and $k' = (1 + \delta)k$ leaves. We show that the cost of the threshold tree $\mathcal{T}$ is at least $\Omega((1 - \delta)k^3)$. We first show that with high probability every two centers in $\{c^1, c^2, \cdots, c^{\tilde{k}+1}\}$ are far apart.

Lemma B.11. *With probability at least $1 - 1/k$ the following statement holds: The distance between every two distinct centers c' and c'' in $\{c^1, c^2, \cdots, c^{\tilde{k}+1}\}$ is at least $k/5$.*

Proof. Consider two distinct centers c', c'' in $\{c^1, c^2, \cdots, c^{\tilde{k}+1}\}$. For every coordinate $j \in \{\tilde{k}, \tilde{k} + 1, \cdots, \tilde{k} + p - 1\}$, the j-th coordinate of centers $\{c^1, c^2, \cdots, c^{\tilde{k}+1}\}$ form a random permutation of

$\{0, 1, \cdots, \tilde{k}\}$. Thus, we have for every $j \in \{\tilde{k}, \tilde{k}+1, \cdots, \tilde{k}+p-1\}$

$$\Pr(|c'_j - c''_j| \geq \frac{\tilde{k}}{2}) = \frac{1}{2}.$$

The distance between c' and c'' is at least $\tilde{k}/2$ with probability $1 - (1/2)^p = 1 - 1/k^3$. By the union bound over all pairs of centers in $\{c^1, c^2, \cdots, c^{\tilde{k}+1}\}$, the distance between two distinct centers in $\{c^1, c^2, \cdots, c^{\tilde{k}+1}\}$ is at least $k/5$ with probability at least $1 - 1/k$. □

By Lemma B.11, we can find an instance X such that the distance between every two distinct centers c' and c'' in $\{c^1, c^2, \cdots, c^{\tilde{k}+1}\}$ is at least $k/5$. Then, we show that there are at least $(1 - 4\delta)\tilde{k}$ data points which are separated from their original centers in the threshold tree $\mathcal{T}$ given by the ExKMC algorithm with the IMM base tree. The algorithm first uses the IMM algorithm in Dasgupta et al. (2020) to generate a threshold tree with k leaves. The IMM algorithm iteratively chooses the threshold cut that minimizes the number of mistakes to separate centers, where a mistake means a data point is separated from its original center.

For this instance X, we show that the first $\tilde{k} - 1$ cuts chosen by the IMM algorithm are at the first $\tilde{k} - 1$ coordinates. At any iteration $t \leq \tilde{k} - 1$, suppose the first $t - 1$ cuts are at the first $\tilde{k} - 1$ coordinates. If any center c^i for $i \in \{2, \cdots, \tilde{k}\}$ is not separated from center c^1, then the threshold cut at coordinate $i - 1$ will separate center c^i from other centers and split one data point at e_j from its center c^1. Note that centers c^1 and $c^{\tilde{k}+2}, c^{\tilde{k}+3}, \ldots, c^k$ are not separated at iteration t. For every coordinate $j \in \{\tilde{k}, \cdots, d\}$, the j-th coordinate of these centers form a permutation of $\{0, 1, \cdots, \tilde{k}\}$. Therefore, every threshold cut at coordinate $j \in \{\tilde{k}, \tilde{k}+1, \cdots, d\}$ will split at least two data points from their centers. Thus, the IMM algorithm will choose a threshold cut at coordinate $i - 1 \leq \tilde{k} - 1$ at iteration t.

We now bound the number of mistakes in the tree $\mathcal{T}$ given by the ExKMC algorithm. Since

the IMM algorithm chooses the first $\tilde{k}-1$ threshold cuts at the first $\tilde{k}-1$ coordinates, the IMM algorithm splits $\tilde{k}-1$ data points at $e_1, e_2, \ldots, e_{\tilde{k}-1}$ from their original center c^1. Since all these $\tilde{k}-1$ data points are separated in $\tilde{k}-1$ leaves of the IMM tree, the ExKMC algorithm with $(1+\delta)k$ leaves can rearrange at most δk data points among these $\tilde{k}-1$ data points to their original centers. Therefore, there are at least $\tilde{k}-1-\delta k \geq (1-4\delta)\tilde{k}$ data points separated from their original centers in the threshold tree $\mathcal{T}$ given by the ExKMC algorithm with the IMM base tree.

By Lemma B.11, the cost of each data point separated from its original center is at least $\Omega(k^2)$. Since $\mathrm{OPT}_k(X) = O(\tilde{k}\log k)$, the cost of the threshold tree $\mathcal{T}$ is at least

$$\mathrm{cost}(X, \mathcal{T}) \geq \Omega((1-4\delta)\tilde{k}\cdot k^2) \geq \Omega((1-4\delta)k^2/\log k)\,\mathrm{OPT}_k(X).$$

□

VITA

Liren Shan was born in Changzhou, a beautiful city located in the northwest of Shanghai. He received a Bachelor of Science in Mathematics and Applied Mathematics from Fudan University in June 2018.