Near-optimal Algorithms for Explainable k-Medians and k-Means[*]

Konstantin Makarychev[†]   Liren Shan[†]

Northwsestern University

Abstract

We consider the problem of explainable k-medians and k-means introduced by Dasgupta, Frost, Moshkovitz, and Rashtchian (ICML 2020). In 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 data based on a single feature into two groups. We propose a new algorithm for this problem which is $\tilde{O}(\log k)$ competitive with k-medians with ℓ_1 norm and $\tilde{O}(k)$ competitive with k-means. This is an improvement over the previous guarantees of $O(k)$ and $O(k^2)$ by Dasgupta et al (2020). We also provide a new algorithm which is $O(\log^{3/2} k)$ competitive for k-medians with ℓ_2 norm. Our first algorithm is near-optimal: Dasgupta et al (2020) showed a lower bound of $\Omega(\log k)$ for k-medians; in this work, we prove a lower bound of $\tilde{\Omega}(k)$ for k-means. We also provide a lower bound of $\Omega(\log k)$ for k-medians with ℓ_2 norm.

1   Introduction

In this paper, 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, we want to find an *explainable* clustering – clustering which can be easily understood by a human being. Then, k-means or k-medians clustering may not be the best options for us.

Note that though 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. Every k-medians or k-means clustering is defined by a set of k centers $c^1, c^2, \dots, c^k$, where each cluster is the set of points located closer to a fixed center c^i than to any other center c^j. That is, for points in cluster i, we must have $\arg\min_j \|x - c^j\| = i$. Thus, 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 every day 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 metres or 150 metres*

---

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.

The above discussion leads us to the idea of Dasgupta et al. (2020), who proposed to use threshold (decision) trees to describe clusters (see also Liu, Xia, and Yu (2005), Fraiman, Ghattas, and Svarc (2013), Bertsimas, Orfanoudaki, and Wiberg (2018), and Saisubramanian, Galhotra, and Zilberstein (2020)).

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 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).

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 T equals (1), the k-medians in ℓ_2 cost equals (2) and k-means cost equals (3):

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

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

$$\text{cost}_{\ell_2^2}(X, T) = \sum_{i=1}^{k} \sum_{x \in P_i} \|x - c^i\|_2^2, \tag{3}$$

where c^i is the ℓ_1-median of cluster P_i in (1), the ℓ_2-median of cluster P_i in (2), and the mean of cluster P_i in (3).

This definition raises obvious questions: Can we actually find a good explainable clustering? Moreover, how good can it be comparing to a regular 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 (regular) k-medians with ℓ_1 norm, k-medians with ℓ_2 norm, and k-means, respectively. Dasgupta et al. (2020) defined the *price of explainability* as the ratio $\text{cost}_{\ell_1}(X, T)/\text{OPT}_{\ell_1}(X)$ 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 best 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. Furthermore, they designed two algorithms that given a k-medians in ℓ_1 or k-means clustering, produce an explainable clustering with cost at most $O(k)$ and $O(k^2)$ times the cost of original clustering (respectively). They also provided examples for which the price of explainability of k-medians in ℓ_1 and k-means is at least $\Theta(\log k)$.

	k-medians in ℓ_1		k-medians in ℓ_2		k-means	
	Lower	**Upper**	**Lower**	**Upper**	**Lower**	**Upper**
Our results		$O(\log k \log \log k)$	$\Omega(\log k)$	$O(\log^{3/2} k)$	$\Omega(k/\log k)$	$O(k \log k \log \log k)$
Dasgupta et al. (2020)	$\Omega(\log k)$	$O(k)$			$\Omega(\log k)$	$O(k^2)$

Figure 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.

1.1 Our results

In this work, we give almost tight bounds on the price of explainability for both k-medians in ℓ_1 and k-means. Specifically, we show how to transform any clustering to an explainable clustering with cost at most $O(\log k \log \log k)$ times the original cost for the k-medians ℓ_1 objective and $O(k \log k \log \log k)$ for the k-means objective. Note that we get an exponential improvement over previous results for the k-medians ℓ_1 objective. Furthermore, we present an algorithm for k-medians in ℓ_2 with the price of explainability bounded by $O(\log^{3/2} k)$. We complement these results with an almost tight lower bound of $\Omega(k/\log k)$ for the k-means objective and an $\Omega(\log k)$ lower bound for k-medians in ℓ_2 objective. We summarise our results in Table 1.

Below, we formally state our main results. The costs of threshold trees and clusterings are defined by formulas (1), (2), (3), (4), (5), and (6).

Theorem 1.1. *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 T with expected k-medians in ℓ_1 cost at most*

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

Theorem 1.2. *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 T with expected k-means cost at most*

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

We note that the algorithms by Dasgupta et al. (2020) also produce trees based on the given set of "reference" centers $c^1, \ldots, c^k$. However, the approximation guarantees of those algorithms are $O(k)$ and $O(k^2)$, respectively. Our upper bound of $O(\log k \log \log k)$ almost matches the lower bound of $\Omega(\log k)$ given by Dasgupta et al. (2020). The upper bound of $O(k \log k \log \log k)$ almost matches the lower bound of $\Omega(k/\log k)$ we show in Section 8.

Theorem 1.3. *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 T with expected k-medians in ℓ_2 cost at most*

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

1.2 Related work

Dasgupta et al. (2020) introduced the *explainable k-medians and k-means clustering problems* and developed Iterative Mistake Minimization (IMM) algorithms for these problems. Later, Frost,

Moshkovitz, and Rashtchian (2020) proposed algorithms that construct threshold trees with more than k leaves.

Decision trees have been used for interpretable classification and clustering since 1980s. Breiman, Friedman, Olshen, and Stone (1984) proposed a popular decision tree algorithm called CART for supervised classification. For unsupervised clustering, threshold decision trees are used in many empirical methods based on different criteria such as information gain Liu et al. (2005), local 1-means cost Fraiman et al. (2013), Silhouette Metric Bertsimas et al. (2018), and interpretability score Saisubramanian et al. (2020).

The k-means and k-medians clustering problems have been extensively studied in the literature. The k-means++ algorithm proposed by Arthur and Vassilvitskii (2006) is the most widely used algorithm for k-means clustering. It provides an $O(\ln k)$ approximation. Li and Svensson (2016) provided a $1 + \sqrt{3} + \varepsilon$ approximation for k-medians in general metric spaces, which was improved to $2.611 + \varepsilon$ by Byrka, Pensyl, Rybicki, Srinivasan, and Trinh (2014). Ahmadian, Norouzi-Fard, Svensson, and Ward (2019) gave a 6.357 approximation algorithm for k-means. The k-medians and k-means problems are NP-hard Megiddo and Supowit (1984); Dasgupta (2008); Aloise et al. (2009). Recently, Awasthi, Charikar, Krishnaswamy, and Sinop (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 et al. (2017)). Bhattacharya, Goyal, and Jaiswal (2020) showed that the Euclidean k-medians can not be approximated within a factor of $(1 + \varepsilon)$ for some constant ε assuming the unique games conjecture.

Boutsidis et al. (2009), Boutsidis et al. (2014), Cohen et al. (2015), Makarychev et al. (2019) and Becchetti et al. (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-medians in ℓ_2 and k-means to $d' = O(\log k)$. Note, however, that the Johnson–Lindenstrauss transform cannot be used for the explainable k-medians and k-means problems, because this transform does not preserve the set of features. Instead, we can use a *feature selection* algorithm by Boutsidis et al. (2014) or Cohen et al. (2015) to reduce the dimensionality to $d' = \tilde{O}(k)$.

Independently of our work, Laber and Murtinho (2021) proposed new algorithms for explainable k-medians with ℓ_1 and k-means objectives. Their competitive ratios are $O(d \log k)$ and $O(dk \log k)$, respectively. Note that these competitive ratios depend on the dimension d of the space.

Remark: After this paper was accepted to ICML 2021, we learned about two independent results that were recently posted on arXiv.org. The first paper by Charikar and Hu (2021) gives a $k^{1-2/d} \text{poly}(d \log k)$-competitive algorithm for k-means. Note that this bound depends on the dimension of the data set. It is better than our k-means bound ($O(k \log k \log \log k)$) for small d ($d \ll \log k / \log \log k$) and worse for large d ($d \gg \log k / \log \log k$). The second paper by Gamlath, Jia, Polak, and Svensson (2021) gives $O(\log^2 k)$ and $O(k \log^2 k)$-competitive algorithms for k-medians in ℓ_1 and k-means, respectively. These bounds are slightly worse than ours.

2 Preliminaries

Given a set of points $X \subseteq \mathbb{R}^d$ and an integer $k > 1$, the regular k-medians and k-means clustering problems are to find a set C of k centers to minimize the corresponding costs: k-medians with ℓ_1

objective cost (4), k-medians with ℓ_2 objective cost (5), and k-means cost (6).

$$\text{cost}_{\ell_1}(X, C) = \sum_{x \in X} \min_{c \in C} \|x_i - c\|_1, \tag{4}$$

$$\text{cost}_{\ell_2}(X, C) = \sum_{x \in X} \min_{c \in C} \|x_i - c\|_2. \tag{5}$$

$$\text{cost}_{\ell_2^2}(X, C) = \sum_{x \in X} \min_{c \in C} \|x_i - c\|_2^2. \tag{6}$$

Every coordinate cut is specified by the coordinate $i \in \{1, \ldots, d\}$ and threshold θ. We denote the set of all possible cuts by Ω:

$$\Omega = \{1, \cdots, d\} \times \mathbb{R}.$$

We define the standard product measure on Ω as follows: The measure of set $S \subset \Omega$ equals

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

where μ_R is the Lebesgue measure on $\mathbb{R}$.

For every cut $\omega = (i, \theta) \in \Omega$ and point $x \in \mathbb{R}^d$, we let

$$\delta_x(\omega) \equiv \delta_x(i, \theta) = \begin{cases} 1, & \text{if } x_i > \theta; \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $\delta_x(i, \theta)$ is the indicator of the event $\{x_i > \theta\}$. Observe that $x \mapsto \delta_x$ is an isometric embedding of ℓ_1^d (d-dimensional ℓ_1 space) into $L_1(\Omega)$ (the space of integrable functions on Ω). Specifically, for $x, y \in \mathbb{R}^d$, we have

$$\begin{aligned} \|x - y\|_1 &\equiv \sum_{i=1}^{d} |x_i - y_i| \\ &= \sum_{i=1}^{d} \int_{-\infty}^{\infty} |\delta_x(i, \theta) - \delta_y(i, \theta)| \, d\theta \\ &= \int_{\Omega} |\delta_x(\omega) - \delta_y(\omega)| \, d\mu(\omega) \equiv \|\delta_x - \delta_y\|_1. \end{aligned} \tag{7}$$

A map $\varphi : \mathbb{R}^d \to \mathbb{R}^d$ is coordinate cut preserving if for every coordinate cut $(i, \theta) \in \Omega$, there exists a coordinate cut $(i', \theta') \in \Omega$ such that $\{x \in \mathbb{R}^d : x_{i'} \le \theta'\} = \{x \in \mathbb{R}^d : \varphi(x)_i \le \theta\}$ and vice versa. In the algorithm for explainable k-means, we use a cut preserving terminal embeddings of "ℓ_2^2 distance" into ℓ_1.

3 Algorithms Overview

We now give an overview of our algorithms.

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 T

Set $S_{ij} = \{\omega \in \Omega : \delta_{c^i}(\omega) \neq \delta_{c^j}(\omega)\}$ for all $i, j \in \{1, \cdots, k\}$. Let $t = 0$.
Create a tree T_0 containing a root vertex r. Assign set $X_r = X \cup C$ to the root.

while T_t contains a leaf with at least two distinct centers c^i and c^j **do**
　　Let $E_t = \bigcup_{\text{leaves } u} \{(i,j) : c^i, c^j \in X_u\}$ be the set of all not yet separated pairs of centers.
　　Let $D_t = \max_{(i,j) \in E_t} \|c^i - c^j\|_1$ be the maximum distance between two not separated centers.

　　Define two sets $A_t, B_t \subset \Omega$ as follows:

$$A_t = \bigcup_{(i,j) \in E_t} S_{ij} \qquad \text{and} \qquad B_t = \bigcup_{\substack{(i,j) \in E_t \\ \text{s.t.} \mu(S_{ij}) \leq D_t/k^3}} S_{ij}.$$

　　Let[1] $R_t = A_t \setminus B_t$. Pick a pair $\omega_t = (i, \theta)$ uniformly at random from R_t.
　　For every leaf node u in T, split the set X_u into two sets:

$$Left = \{x \in X_u : x_i \leq \theta\} \qquad \text{and} \qquad Right = \{x \in X_u : x_i > \theta\}.$$

　　If each of these sets contains at least one center from C, then create two children of u in tree T and assign sets $Left$ and $Right$ to the left and right child, respectively.

　　Denote the updated tree by T_{t+1}.
　　Update $t = t + 1$.
end while

Figure 2: Threshold tree construction for k-medians in ℓ_1

k-medians in ℓ_1. We begin with the algorithm for k-medians in ℓ_1. We show that its competitive ratio is $O(\log^2 k)$ in Section 4 and then show an improved bound of $O(\log k \log \log k)$ in Section 5.

As the algorithm by Dasgupta et al. (2020), our algorithm (see Algorithm 2) builds a binary threshold tree T top-down. It starts with a tree containing only the root node r. This node is assigned the set of points X_r that contains all points in the data set X and all reference centers c^i. At every round, the algorithm picks some pair $\omega = (i, \theta) \in \Omega$ (as we discuss below) and then splits data points x assigned to every *leaf* node u into two groups $\{x \in X_u : x_i \leq \theta\}$ and $\{x \in X_u : x_i > \theta\}$. Here, X_u denotes the set of points assigned to the node u. If this partition separates at least two centers c^i and c^j, then the algorithm attaches two children to u and assigns the first group to the left child and the second group to the right child. The algorithm terminates when all leaves contain exactly one reference center c^i. Then, we assign the points in each leaf of T to its unique reference center. Note that the unique reference center in each leaf may not be the optimal center for points contained in that leaf. Thus, the total cost by assigning each point to the

reference center in the same leaf of T is an upper bound of the cost of threshold tree T.

The algorithm by Dasgupta et al. (2020) picks splitting cuts in a greedy way. Our algorithm chooses them at random. Specifically, to pick a cut $\omega_t \in \Omega$ at round t, our algorithm finds the maximum distance D_t between two distinct centers c^i, c^j that belong to the same set X_u assigned to a leaf node u i.e.,

$$D_t = \max_{u \text{ is a leaf}} \max_{c^i, c^j \in X_u} \|c^i - c^j\|_1.$$

Then, we let A_t be the set of all $\omega \in \Omega$ that separate at least one pair of centers; and B_t be the set of all $\omega \in \Omega$ that separate two centers at distance at most D_t/k^3. We pick ω_t uniformly at random (with respect to measure μ) from the set $R_t = A_t \setminus B_t$.

Every $\omega \in R_t$ is contained in A_t, which means ω separates at least one pair of centers. Thus, our algorithm terminates in at most $k-1$ iterations. It is easy to see that the running time of this algorithm is polynomial in the number of clusters k and dimension of the space d. In Section 9, we provide a variant of this algorithm with running time $\tilde{O}(kd)$.

k**-medians in ℓ_2.** Our algorithm for k-medians with ℓ_2 norm recursively partitions the data set X using the following idea. It finds the median point m of all centers in X. Then, it repeatedly makes cuts that separate centers from m. To make a cut, the algorithm chooses a random coordinate $i \in \{1, \ldots, d\}$, random number $\theta \in [0, R^2]$, and random sign $\sigma \in \{\pm 1\}$, where R is the largest distance from a center in X to the median point m. It then makes a threshold cut $(i, m_i + \sigma\sqrt{\theta})$. After separating more than half centers from m, the algorithm recursively calls itself for each of the obtained parts. In Section 7, we show that the *price of explainability* for this algorithm is $O(\log^{3/2} k)$.

k**-means.** We now move to the algorithm for k-means. This algorithm embeds the space ℓ_2 into ℓ_1 using a specially crafted *terminal embedding* φ (the notion of terminal embeddings was formally defined by Elkin et al. (2017)). The embedding satisfies the following property for every center c (terminal) and every point $x \in \ell_2$, we have

$$\|\varphi(x) - \varphi(c)\|_1 \leq \|x - c\|_2^2 \leq 8k \cdot \|\varphi(x) - \varphi(c)\|_1.$$

Then, the algorithm partitions the data set $\varphi(X)$ with centers $\varphi(c^1), \ldots, \varphi(c^k)$ using Algorithm 2. The expected cost of partitioning is at most the distortion of the embedding $(8k)$ times the competitive guarantee $(O(\log k \log\log k))$ of Algorithm 2. In Section 8, we show an almost matching lower bound of $\Omega(k/\log k)$ on the cost of explainability for k-means. We also remark that the terminal embedding we use in this algorithm cannot be improved. This follows from the fact that the cost function $\|x - c\|_2^2$ does not satisfy the triangle inequality; while the ℓ_1 distance $\|\varphi(x) - \varphi(c)\|_1$ does.

4   Algorithm for k-medians in ℓ_1

In this section, we analyse Algorithm 2 for k-medians in ℓ_1 and show that it provides an explainable clustering with cost at most $O(\log^2 k)$ times the original cost. We improve this bound to $O(\log k \log\log k)$ in Section 5.

Recall, all centers in C are separated by the tree T returned by the algorithm, and each leaf of T contains exactly one center from C. For each point $x \in X$, we define its cost in the threshold

---

[1]As we discuss in Section 9, we can also let $R_t = A_t$. However, this change will make the analysis of the algorithm a little more involved.

tree T as
$$\mathrm{alg}_{\ell_1}(x) = \|x - c\|_1,$$
where c is the center in the same leaf in T as x. Then, $\mathrm{cost}_{\ell_1}(X, T) \le \sum_{x \in X} \mathrm{alg}_{\ell_1}(x)$ (note that the original centers $c^1, \ldots, c^k$ used in the definition of $\mathrm{alg}_{\ell_1}(x)$ are not necessarily optimal for the tree T. Hence, the left hand side is not always equal to the right hand side.). For every point $x \in X$, we also define $\mathrm{cost}_{\ell_1}(x, C) = \min_{c \in C} \|x - c\|_1$. Then, $\mathrm{cost}_{\ell_1}(X, C) = \sum_{x \in X} \mathrm{cost}_{\ell_1}(x, C)$ (see (4)).

We prove the following theorem.

Theorem 4.1. *Given a set of points X in $\mathbb{R}^d$ and a set of centers $C = \{c^1, \ldots, c^k\} \subset \mathbb{R}^d$, Algorithm 2 finds a threshold tree T with expected k-medians in ℓ_1 cost at most*
$$\mathbb{E}[\mathrm{cost}_{\ell_1}(X, T)] \le O(\log^2 k) \cdot \mathrm{cost}_{\ell_1}(X, C).$$
Moreover, the same bound holds for the cost of every point $x \in X$ i.e.,
$$\mathbb{E}[\mathrm{cost}_{\ell_1}(x, T)] \le O(\log^2 k) \cdot \mathrm{cost}_{\ell_1}(x, C).$$

Proof. Let T_t be the threshold tree constructed by Algorithm 2 before iteration t. Consider a point x in X. If x is separated from its original center in C by the cut generated at iteration t, then x will be eventually assigned to some other center in the same leaf of T_t. By the triangle inequality, the new cost of x at the end of the algorithm will be at most $\mathrm{cost}_{\ell_1}(x, C) + D_t$, where D_t is the maximum diameter of any leaf in T_t (see Algorithm 2). Define a penalty function $\phi_t(x)$ as follows: $\phi_t(x) = D_t$ if x is separated from its original center c at time t; $\phi_t(x) = 0$, otherwise. Note that $\phi_t(x) \ne 0$ for at most one iteration t, and
$$\mathrm{alg}_{\ell_1}(x) \le \mathrm{cost}_{\ell_1}(x, C) + \sum_t \phi_t(x). \tag{8}$$

The sum in the right hand side is over all iterations of the algorithm. We bound the expected penalty $\phi_t(x)$ for each t.

Lemma 4.2. *The expected penalty $\phi_t(x)$ is upper bounded as follows:*
$$\mathbb{E}[\phi_t(x)] \le \mathbb{E}\left[D_t \cdot \int_\Omega |\delta_x(\omega) - \delta_c(\omega)| \cdot \frac{\mathbb{1}\{\omega \in R_t\}}{\mu(R_t)} \, d\mu(\omega)\right],$$
where c is the closest center to the point x in C; $\mathbb{1}\{\omega \in R_t\}$ is the indicator of the event $\omega \in R_t$.

Proof. If x is already separated from its original center c at iteration t, then $\phi_t(x) = 0$. Otherwise, x and c are separated at iteration t if for the random pair $\omega_t = (i, \theta)$ chosen from R_t in Algorithm 2, we have $\delta_x(\omega_t) \ne \delta_c(\omega_t)$. Write,
$$\mathbb{E}[\phi_t(x)] \le \mathbb{E}\left[\mathbb{P}_{\omega_t}[\delta_x(\omega_t) \ne \delta_c(\omega_t) \mid T_t] \cdot D_t\right].$$

The probability that $\delta_x(\omega_t) \ne \delta_c(\omega_t)$ given T_t is bounded as
$$\mathbb{P}_{\omega_t}[\delta_x(\omega_t) \ne \delta_c(\omega_t) \mid T_t] = \frac{\mu\{\omega \in R_t : \delta_x(\omega) \ne \delta_c(\omega)\}}{\mu(R_t)}$$
$$= \int_\Omega \mathbb{1}\{\delta_x(\omega) \ne \delta_c(\omega)\} \cdot \frac{\mathbb{1}\{\omega \in R_t\}}{\mu(R_t)} \, d\mu(\omega)$$
$$= \int_\Omega |\delta_x(\omega) - \delta_c(\omega)| \cdot \frac{\mathbb{1}\{\omega \in R_t\}}{\mu(R_t)} \, d\mu(\omega).$$

$\square$

Let

$$W_t(\omega) = \frac{D_t \cdot \mathbb{1}\{\omega \in R_t\}}{\mu(R_t)}.$$

Then, by Lemma 4.2 and inequality (8), we have

$$\mathbb{E}[\mathrm{alg}_{\ell_1}(x)] \leq \mathrm{cost}_{\ell_1}(x, C) + \mathbb{E}\Big[\sum_t \int_\Omega |\delta_x(\omega) - \delta_c(\omega)| \cdot W_t(\omega) \, d\mu(\omega) \Big].$$

The upper bound on the expected cost of x in tree T consists of two terms: The first term is the original cost of x. The second term is a bound on the expected penalty incurred by x. We now bound the second term as $O(\log^2 k) \cdot \mathrm{cost}_{\ell_1}(x, C)$.

$$\mathbb{E}\Big[\sum_t \int_\Omega |\delta_x(\omega) - \delta_c(\omega)| \cdot W_t(\omega) \, d\mu(\omega) \Big] = \int_\Omega |\delta_x(\omega) - \delta_c(\omega)| \cdot \mathbb{E}\Big[\sum_t W_t(\omega) \Big] d\mu(\omega).$$

By Hölder's inequality, the right hand side is upper bounded by the following product:

$$\|\delta_x - \delta_c\|_1 \cdot \max_{\omega \in \Omega} \mathbb{E}\Big[\sum_t W_t(\omega) \Big].$$

The first multiplier in the product exactly equals $\|x - c\|_1$ (see Equation 7), which, in turn, equals $\mathrm{cost}_{\ell_1}(x, C)$. Hence, to finish the proof of Theorem 4.1, we need to upper bound the second multiplier by $O(\log^2 k)$.

Lemma 4.3. *For all $\omega \in \Omega$, we have*

$$\mathbb{E}\Big[\sum_t W_t(\omega) \Big] \leq O(\log^2 k).$$

Proof. Let t' be the first iteration and t'' be the last iteration for which $W_t(\omega) > 0$. First, we prove that $D_{t''} \geq D_{t'}/k^3$, where $D_{t'}$ and $D_{t''}$ are the maximum cluster diameters at iterations t' and t'', respectively. Since $W_{t'}(\omega) > 0$ and $W_{t''}(\omega) > 0$, we have $\mathbb{1}\{\omega \in R_{t'}\} \neq 0$ and $\mathbb{1}\{\omega \in R_{t''}\} \neq 0$. Hence, $\omega \in R_{t'}$ and $\omega \in R_{t''}$. Since $\omega \in R_{t''}$, there exists a pair $(i, j) \in E_{t''}$ for which $\omega \in S_{ij}$. For this pair, we have $D_{t''} \geq \mu(S_{ij})$. Observe that the pair (i, j) also belongs to $E_{t'}$, since $E_{t''} \subset E_{t'}$. Moreover, $\mu(S_{ij}) > D_{t'}/k^3$, because otherwise, S_{ij} would be included in $B_{t'}$ (see Algorithm 2) and, consequently, ω would not belong to $R_{t'} = A_{t'} \setminus B_{t'}$. Thus

$$D_{t''} \geq \mu(S_{ij}) > D_{t'}/k^3. \tag{9}$$

By the definition of t' and t'', we have

$$\sum_t W_t(\omega) = \sum_{t=t'}^{t''} W_t(\omega) \leq \sum_{t=t'}^{t''} \frac{D_t}{\mu(R_t)}.$$

Note that the largest distance D_t is a non-increasing (random) function of t. Thus, we can split the iterations of the algorithm $\{t', ..., t''\}$ into $\lceil 3 \log k \rceil$ phases. At phase s, the maximum diameter D_t is in the range $(D_{t'}/2^{s+1}, D_{t'}/2^s]$. Denote the set of all iterations in phase s by $\mathrm{Phase}(s)$.

Consider phase s. Let $D = D_{t'}/2^s$. Phase s ends when all sets S_{ij} with $\mu(S_{ij}) \geq D/2$ are removed from the set E_t. Let us estimate the probability that one such set S_{ij} is removed from

9

E_t at iteration t. Set S_{ij} is removed from E_t if the random threshold cut ω_t chosen at iteration t separates centers c_i and c_j, or, in other words, if $w_t \in S_{ij}$. The probability of this event equals:

$$\mathbb{P}[\omega_t \in S_{ij} \mid T_t] = \frac{\mu(S_{ij} \cap R_t)}{\mu(R_t)} = \frac{\mu(S_{ij}) - \mu(S_{ij} \cap B_t)}{\mu(R_t)} \geq \frac{\mu(S_{ij}) - \mu(B_t)}{\mu(R_t)}.$$

Note that $\mu(S_{ij}) > D/2 \geq D_t/2$ and $\mu(B_t) < \binom{k}{2} \cdot \frac{D_t}{k^3} < \frac{D_t}{2k}$ (because B_t is the union of at most $\binom{k}{2}$ sets of measure at most D_t/k^3 each). Hence,

$$\mathbb{P}[\omega_t \in S_{ij} \mid T_t] \geq \frac{D_t}{4\mu(R_t)} \geq \frac{1}{4}W_t(\omega).$$

If $W_t(\omega)$ did not depend on t, then we would argue that each set S_{ij} (with $\mu(S_{ij}) \geq D/2$) is removed from E_t in at most $4/W_t(\omega)$ iterations, in expectation, and, consequently, all sets S_{ij} are removed in at most $O(\log k) \cdot 4/W_t(\omega)$ iterations, in expectation (note that the number of sets S_{ij} is upper bounded by $\binom{k}{2}$). Therefore,

$$\mathbb{E}\Big[\sum_{t \in \text{Phase}(s)} W_t(\omega)\Big] \leq O(\log k) \cdot \frac{4}{W_t(\omega)} \cdot W_t(\omega) = O(\log k).$$

However, we cannot assume that $W_t(\omega)$ is a constant. Instead, we use the following claim with $E = \{0, \ldots, k-1\} \times \{0, \ldots, k-1\}$, $E'_t = \{(i,j) \in E_t : \mu(S_{ij}) \geq D/2\}$, and $p_t = W_t(\omega)/4$.

Claim 4.4. *Consider two stochastic processes E_t and p_t adapted to filtration $\mathcal{F}_t$. The values of E_t are subsets of some finite non-empty set E. The values of p_t are numbers in $[0,1]$. Suppose that for every step t, $E_{t+1} \subset E_t$ and for every $e \in E_t$, $\Pr[e \notin E_{t+1} \mid \mathcal{F}_t] \geq p_t$. Let τ be the (stopping) time t when $E_t = \varnothing$. Then,*

$$\mathbb{E}\Big[\sum_{t=0}^{\tau-1} p_t\Big] \leq \ln|E| + O(1).$$

Proof of Claim 4.4. Let τ_e be the first time $t \geq 1$ when element e does not belong to E_t. Then, $\tau = \max_{e \in E} \tau_e$. Hence,

$$\sum_{t=0}^{\tau-1} p_t = \max_{e \in E} \sum_{t=0}^{\tau_e-1} p_t.$$

By the union bound, for all $\lambda \geq 0$, we have

$$\mathbb{P}\Big[\sum_{t=0}^{\tau-1} p_t \geq \lambda\Big] \leq \sum_{e \in E} \mathbb{P}\Big[\sum_{t=0}^{\tau_e-1} p_t \geq \lambda\Big]. \tag{10}$$

Define a new stochastic process $Z_t(e)$ as follows: $Z_0(e) = 1$ and for $t \geq 1$,

$$Z_t(e) = \begin{cases} e^{\sum_{t'=0}^{t-1} p_{t'}}, & \text{if } e \in E_t; \\ 0, & \text{otherwise.} \end{cases}$$

Note that if $\sum_{t=0}^{\tau_e-1} p_t \geq \lambda$, then $\max_{t \geq 0} Z_t(e) \geq e^{\lambda-1}$. Thus, we will bound $\Pr[\max_{t \geq 0} Z_t(e) \geq e^{\lambda-1}]$. Observe that Z_t is a supermartingale, since

$$\mathbb{E}[Z_{t+1} \mid \mathcal{F}_t] = \Pr[e \in E_{t+1} \mid \mathcal{F}_t] \cdot e^{p_t} \cdot Z_t$$
$$\leq (1 - p_t) \cdot e^{p_t} \cdot Z_t \leq Z_t.$$

10

By Doob's maximal martingale inequality, we have

$$\Pr[\max_{t \geq 0} Z_t(e) \geq e^{\lambda - 1}] \leq Z_0(e)/e^{\lambda - 1} = e^{-(\lambda - 1)}.$$

Using (10), we get

$$\mathbb{P}\left[\sum_{t=0}^{\tau - 1} p_t \geq \lambda\right] \leq |E| \cdot e^{-(\lambda - 1)}.$$

Therefore,

$$\mathbb{E}\left[\sum_{t=0}^{\tau - 1} p_t\right] = \int_0^\infty \mathbb{P}\left[\sum_{t=0}^{\tau - 1} p_t \geq \lambda\right] d\lambda \leq \ln |E| + \int_{\ln |E|}^\infty |E| \cdot e^{-(\lambda - 1)} d\lambda$$

$$= \ln |E| + |E| e^{-\ln |E| + 1} = \ln |E| + e.$$

$\square$

By Claim 4.4,

$$\mathbb{E}\left[\sum_{t \in \text{Phase}(s)} W_t(\omega)\right] \leq O(\log k).$$

The expected sum of W_t over all phases is upper bounded by $O(\log^2 k)$, since the number of phases is upper bounded by $O(\log k)$. We note that if the number of phases is upper bounded by L, then the expected sum of W_t over all phases is upper bounded by $O(L \log k)$. This concludes the proofs of Lemma 4.3 and Theorem 4.1. $\square$

5 Improved Analysis for k-medians in ℓ_1

In this section, we provide an improved analysis of our algorithm for k-medians in ℓ_1.

Theorem 5.1. *Given a set of points X in $\mathbb{R}^d$ and set of centers $C = \{c^1, \ldots, c^k\} \subset \mathbb{R}^d$, Algorithm 2 finds a threshold tree T with expected k-medians ℓ_1 cost at most*

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

Proof. In the proof of Theorem 4.1, we used a pessimistic estimate on the penalty a point $x \in X$ incurs when it is separated from its original center c. Specifically, we bounded the penalty by the maximum diameter of any leaf in the tree T_t. In the current proof, we will use an additional bound: The distance from x to the closest center after separation. Suppose, that x is separated from its original center c. Let c' be the closest center to x after we make cut ω_t at step t. That is, c' is the closest center to x in the same leaf of the threshold tree T_{t+1}. Note that after we make additional cuts, x may be separated from its new center c' as well, and the cost of x may increase. However, as we already know, the expected cost of x may increase in at most $O(\log^2 k)$ times in expectation (by Theorem 4.1). Here, we formally apply Theorem 4.1 to the leaf where x is located and treat c' as the original center of x. Therefore, if x is separated from c by a cut ω_t at step t, then the expected cost of x in the end of the algorithm is upper bounded by

$$\mathbb{E}[\text{alg}_{\ell_1}(x) \mid T_t, \omega_t] \leq O(\log^2 k) \cdot \|c' - x\|_1 = O(\log^2 k) \cdot D_t^{min}(x, \omega_t).$$

11

In the formula above, we used the following definition: $D_t^{min}(x,\omega)$ is the distance from x to the closest center c' in the same leaf of T_t as x which is not separated from x by the cut ω i.e., $\delta_x(\omega) = \delta_{c'}(\omega)$. If there are no such centers c' (i.e., cut ω separates x from all centers), then we let $D_t^{min}(x,\omega) = 0$. Note that in this case, our algorithm will never make cut ω, since it always makes sure that the both parts of the cut contain at least one center from C. Similarly to $D_t^{min}(x,\omega)$, we define $D_t^{max}(x,\omega)$: $D_t^{max}(x,\omega)$ is the distance from x to the farthest center c'' in the same leaf of T_t as x which is not separated from x by the cut ω. We also let $D_t^{max}(x,\omega) = 0$ if there is no such c''. Note that $D_t^{max}(x,\omega)$ is an upper bound on the cost of x in the eventual threshold tree T if cut ω separated x from c at step t.

We now have three bounds on the expected cost of x in the final tree T given that the algorithm separates x from its original center c at step t with cut ω. The first bound is $D_t^{max}(x,\omega)$; the second bound is $O(\log^2 k) \cdot D_t^{min}(x,\omega)$, and the third bound is $\|x - c\|_1 + D_t$. We use the first bound if $D_t^{max}(x,\omega) \leq 2\|x - c\|_1$. We call such cuts ω light cuts. We use the second bound if $D_t^{max}(x,\omega) > 2\|x - c\|_1$ but $D_t^{min}(x,\omega) \leq D_t/\log^4 k$. We call such cuts ω medium cuts. We use the third bound if $D_t^{max}(x,\omega) > 2\|x - c\|_1$ and $D_t^{min}(x,\omega) > D_t/\log^4 k$. We call such cuts ω heavy cuts.

Note that in the threshold tree returned by the algorithm, one and only one of the following may occur: (1) x is separated from the original center c by a light, medium, or heavy cut; (2) x is not separated from c. We now estimate expected penalties due to light, medium, or heavy cuts.

If the algorithm makes a light cut, then the maximum cost of point x in T is at most $2\|x - c\|_1 = 2\text{cost}_{\ell_1}(x, C)$. So we should not worry about such cuts. If the algorithm makes a medium cut, then the expected additional penalty for x is upper bounded by

$$D_t^{min}(x,\omega_t) \cdot O(\log^2 k) \leq O(\phi_t(x)/\log^2 k),$$

where $\phi_t(x)$ is the function from the proof of Theorem 4.1. Thus, the total expected penalty due to a medium cut (added up over all steps of the algorithm) is $\Omega(\log^2 k)$ times smaller than the penalty we computed in the proof of Theorem 4.1. Therefore, the expected penalty due to a medium cut is at most $O(\|x - c\|_1)$.

We now move to heavy cuts. Denote the set of possible heavy cuts for x in R_t by H_t. That is, if x is not separated from its original center c by step t, then

$$H_t = \left\{ \omega \in R_t : \ D_t^{min}(x,\omega) > D_t/\log^4 k \text{ and } D_t^{max}(x,\omega) > 2\|x - c\|_1 \right\}.$$

Otherwise, let $H_t = \varnothing$. Define a density function $\widetilde{W}_t(\omega)$ similarly to $W_t(\omega)$:

$$\widetilde{W}_t(\omega) = \frac{D_t \cdot \mathbb{1}\{\omega \in H_t\}}{\mu(R_t)}.$$

Then, the expected penalty due to a heavy cut is bounded, similarly to Lemma 4.2, by

$$\sum_t \mathbb{E}\left[\int_0^1 |\delta_x(\omega) - \delta_c(\omega)| \cdot \widetilde{W}_t \, d\mu(\omega) \right].$$

Therefore, to finish the proof of Theorem 1.1, we need to prove the following analog of Lemma 4.3.

Lemma 5.2. *For all $\omega \in \Omega$, we have*

$$\mathbb{E}\left[\sum_t \widetilde{W}_t(\omega) \right] \leq O(\log k \log \log k).$$

Proof. As in the proof of Lemma 4.3, consider the first and last steps when $\widetilde{W}_t(\omega) > 0$. Denote these steps by t^* and t^{**}, respectively. In the proof of Lemma 4.3, we had a bound $D_{t''} \geq D_{t'}/k^3$ (see inequality (9)). We now show a stronger bound on t^* and t^{**}.

Claim 5.3. *We have $D_{t^{**}} \geq {}^{D_{t^*}}/_{2\log^4 k}$.*

This claim implies that the number of phases defined in Lemma 4.3 is bounded by $O(\log\log k)$, which immediately implies Lemma 5.2. So, to complete the proof, it remains to show Claim 5.3.

Proof of Claim 5.3 First, note that $\mathbb{1}\{\omega \in H_{t^{**}}\} > 0$ and, consequently, cut ω is heavy at step t^{**}. Thus, $D_{t^{**}}^{min}(x, \omega)$ is positive. Hence, this cut separates c from at least one other center c' in the same leaf of the current threshold tree $T_{t^{**}}$. Let c'' be the farthest such center from point x. Then, $\|c'' - x\|_1 = D_{t^{**}}^{max}(x, \omega)$. Since centers c and c'' are not separated prior to step t^{**}, we have

$$D_{t^{**}} \geq \|c - c''\|_1 \geq \|x - c''\|_1 - \|x - c\|_1.$$

Since ω is a heavy cut and not a light cut, $\|x - c''\|_1 > 2\|x - c\|_1$. Thus,

$$D_{t^{**}} \geq \frac{\|x - c''\|_1}{2} = \frac{D_{t^{**}}^{max}(x, \omega)}{2} \geq \frac{D_{t^{**}}^{min}(x, \omega)}{2}.$$

Now, observe that the random process $D_{t^{**}}^{min}(x, \omega)$ is non-decreasing (for fixed x and ω) since the distance from x to the closest center c' cannot decrease over time. Therefore,

$$D_{t^{**}} \geq \frac{D_{t^{**}}^{min}(x, \omega)}{2} \geq \frac{D_{t^*}^{min}(x, \omega)}{2} \geq \frac{D_{t^*}}{2\log^4 k}.$$

In the last inequality, we used that ω is a heavy cut at time t^*. This finishes the proof of Claim 5.3. $\qquad\square$

6 Terminal Embedding of ℓ_2^2 into ℓ_1

In this section, 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 6.1. *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 6.2. *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*

$$|\psi_K(x) - \psi_K(y)| \leq |x - y|^2 \leq 8|K| \cdot |\psi_K(x) - \psi_K(y)|. \tag{11}$$

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 $\varepsilon_x = \text{sign}(x - y_i)$. 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. Then, we show that this function ψ_K is a cut preserving embedding satisfying inequality (11).

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 $\mathbb{R}$. 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 $|x - y^*| \leq |x - y| = \left((x - y)^2\right)'_x / 2$. 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

$$(j - i) \cdot \sum_{m=i}^{j-1} (y_{m+1} - y_m)^2 \geq \left(\sum_{m=i}^{j-1} y_{m+1} - y_m \right)^2 = (y_j - y_i)^2.$$
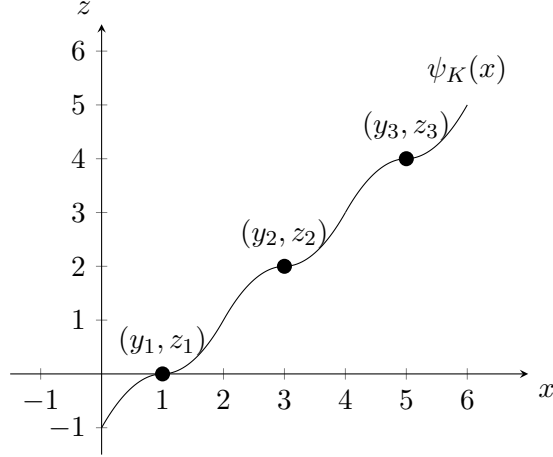
14

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

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 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. $\qquad\square$

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)

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

15

Since ψ_i is a cut preserving terminal embedding by Lemma 6.2, 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. $\qquad\square$

For explainable k-means clustering, we first use the terminal embedding of ℓ_2^2 into ℓ_1. Then, we apply Algorithm 2 to the instance after the embedding. By using this terminal embedding, we can get the following result.

Theorem 6.3. *Given a set of points X in $\mathbb{R}^d$ and a set of centers C in $\mathbb{R}^d$, Algorithm 2 with terminal embedding finds a threshold tree T with expected k-means cost at most*

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

Proof. Let φ be the terminal embedding of ℓ_2^2 into ℓ_1 with terminals C. Let T' be the threshold tree returned by our algorithm on the instance after embedding. Since the terminal embedding φ is coordinate cut preserving, the threshold tree T' also provides a threshold tree 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

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

where the first and third inequality is from the terminal embedding in Lemma 6.1 and the second inequality is due to Theorem 5.1. $\qquad\square$

7 k-medians in ℓ_2

In this section, we present an algorithm for the k-medians in ℓ_2 and show that it provides an explainable clustering with cost at most $O(\log^{3/2} k)$ times the original cost.

7.1   Algorithm for k-medians in ℓ_2

Our algorithm builds a binary threshold tree T using a top-down approach, as shown in Algorithm 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

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 T

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

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

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 4: Threshold tree construction for k-medians in L_2

m_i^u) is a median for i-th coordinates of centers in X_u. That is, for each coordinate i, both sets $\{c \in X_u \cap C : c_i < m_i^u\}$ 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.

Theorem 7.1. *Given a set of points X in $\mathbb{R}^d$ and a set of centers $C = \{c^1, \ldots, c^k\} \subset \mathbb{R}^d$, Algorithm 4 finds a threshold tree T with expected k-medians in ℓ_2 cost at most*

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

Proof. Let $T_t(u)$ 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 $T_t(u)$ as x. That is, if x belongs to a leaf node v in the threshold tree $T_t(u)$, then

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

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 $T_t(u)$. 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 $T_t(u)$ 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

$$\mathrm{cost}_{\ell_2}(x, T) \leq \mathrm{cost}_{\ell_2}(x, C) + \sum_{u \in U_x} \sum_t \phi_t^u(x). \tag{12}$$

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 larger as $k-1$). This means that the size of U_x is at most $O(\log k)$. In Lemma 7.3, 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 7.2. *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,

$$\mathbb{P}\left[\delta_x(i, \vartheta_t^u) \neq \delta_c(i, \vartheta_t^u) \mid T_t(u)\right] = \frac{\left|(c_i - m_i^u)^2 - (x_i - m_i^u)^2\right|}{2(R_t^u)^2}$$

$$\leq \frac{|c_i - x_i|\left(|c_i - m_i^u| + |x_i - m_i^u|\right)}{2(R_t^u)^2},$$

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 $(x_i - m_i^u)(c_i - m_i^u) < 0$,

$$\mathbb{P}\left[\delta_x(i, \vartheta_t^u) \neq \delta_c(i, \vartheta_t^u) \mid T_t(u)\right] = \frac{(c_i - m_i^u)^2 + (x_i - m_i^u)^2}{2(R_t^u)^2}$$

$$\leq \frac{|c_i - x_i|\left(|c_i - m_i^u| + |x_i - m_i^u|\right)}{2(R_t^u)^2},$$

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

$$\mathbb{P}[\delta_x(i_t^u, \vartheta_t^u) \neq \delta_c(i_t^u, \vartheta_t^u) \mid T_t(u)] \leq \sum_{i=1}^{d} \frac{|c_i - x_i|\left(|c_i - m_i^u| + |x_i - m_i^u|\right)}{2d \cdot (R_t^u)^2}$$

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

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

$$\mathbb{E}[\phi_t^u(x)] \leq \mathbb{E}\left[\mathbb{P}\left[\delta_x(i_t^u, \vartheta_t^u) \neq \delta_c(i_t^u, \vartheta_t^u) \mid T_t(u)\right] \cdot 2R_t^u\right]$$

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

$\square$

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 called it a heavy cut.

Lemma 7.3. *In every call of* PARTITION_LEAF(u) *(see Algorithm 4), the expected penalty for a point $x \in X$ is upper bounded as follows:*

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

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 $O(\sqrt{\log k})\text{cost}_{\ell_2}(x, C)$.

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 7.2, 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

$$\mathbb{P}\left[\delta_{c'}(i_t^u, \vartheta_t^u) \neq \delta_{m^u}(i_t^u, \vartheta_t^u) \mid T_t(u)\right] = \sum_{j=1}^d \frac{1}{d} \cdot \frac{(c_j' - m_j^u)^2}{2(R_t^u)^2} = \frac{\|c' - m^u\|_2^2}{2d \cdot (R_t^u)^2} \geq \frac{1}{4d},$$

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

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

$$\leq O(\log k) \cdot \|x - c\|_2 \cdot \frac{\|x - m^u\|_2 + \|c - m^u\|_2}{R}.$$

Let s' be the last phase for which

$$R_1^u/2^{s'} \geq \sqrt{\log_2 k} \cdot \max\{\|x - m^u\|_2, \|c - m^u\|_2\}. \tag{13}$$

20

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

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

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

Using the definition (13) of s', we write

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

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

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 7.2, 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

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

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

This concludes the proof of Lemma 7.3. $\qquad\square$

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 7.3 and Equation (12), we get the conclusion. $\qquad\square$

8 Lower Bound for Threshold Tree

8.1 Lower bound for k-means

In this section, we show a lower bound on the price of explainability for k-means.

Theorem 8.1. *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 8.2. *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

$$\mathbb{P}\left[\sum_{i=1}^{d} X_i - \mathbb{E}\left[\sum_{i=1}^{d} X_i\right] \leq -\sqrt{2d\ln k}\right] \leq e^{-4\ln k} = \frac{1}{k^4},$$

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 $1 - 1/k^2$. $\square$

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 8.3. *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 $\varepsilon \leq \mathbb{P}\left[Y_i^j = 1\right] \leq 2\varepsilon$. Since random variables Y_i^j are independent, by the Chernoff bound for Bernoulli random variables, we have

$$\mathbb{P}\left[\sum_j Y_i^j - \mathbb{E}\left[\sum_j Y_i^j\right] \leq -\sqrt{18\varepsilon k \ln k} \mid \theta\right] \leq e^{-4\ln k} = \frac{1}{k^4}.$$

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. $\square$

Proof of Theorem 8.1. By Lemma 8.2, 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 8.3, 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

$$\text{cost}_{\ell_2^2}(X, T) \geq \frac{\varepsilon k}{4} \cdot \frac{d}{20} = \frac{\varepsilon k d}{80}.$$

Note that the optimal regular k-means cost for this instance X is

$$\text{OPT}_{\ell_2^2}(X) = 2k \cdot \varepsilon^2 d.$$

Therefore, the k-means cost for this instance X given by any threshold tree T is at least

$$\text{cost}_{\ell_2^2}(X, T) \geq \frac{1}{160\varepsilon} \cdot \text{OPT}_{\ell_2^2}(X) = \Omega\left(\frac{k}{\log k}\right) \cdot \text{OPT}_{\ell_2^2}(X).$$

□

8.2 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 8.4. *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*

$$\text{cost}_{\ell_2}(X, T) \geq \Omega(\log k)\text{OPT}_{\ell_2}(X).$$

To prove this lower bound, we use the construction similar to that used in Theorem 8.1. 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 8.1, we also add many points at each center such that the optimal centers for any threshold tree remain almost the same.

Similar to Lemma 8.2, we show that any two centers defined above are far apart with high probability.

Lemma 8.5. *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 8.2, 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 8.6. *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

$$\mathbb{E}\left[\sum_j X_j\right] \geq \varepsilon\left|F_{u(\pi)}\right|.$$

24

Let $\mu = \mathbb{E}[\sum_j X_j]$. By the Chernoff bound for Bernoulli random variables, we have

$$\mathbb{P}\left[\sum_j X_j \le \varepsilon \left|F_{u(\pi)}\right|/2\right] \le \mathbb{P}\left[\sum_j X_j \le \mu/2\right] \le e^{-\mu/8} \le e^{-\varepsilon\sqrt{k}/8}.$$

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 $m(2d/\varepsilon)^m \le e^{-\log^2 k}$. Thus, by the union bound over all paths with length less than $\log_2 k/4$, we get the conclusion. $\qquad\square$

Proof of Theorem 8.4. By Lemma 8.5 and Lemma 8.6, 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

$$\text{cost}_{\ell_2}(X,T) \ge \frac{k}{2}\cdot\frac{\sqrt{d}}{8} = \frac{k\sqrt{d}}{16}.$$

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 8.6, 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

$$\text{cost}_{\ell_2}(X,T) \ge \frac{\lfloor\log_2 k\rfloor}{4}\frac{\varepsilon k}{8}\frac{\sqrt{d}}{8} = \Omega(k\sqrt{d}\varepsilon\log k).$$

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

$$\text{cost}_{\ell_2}(X,T) = \Omega(k\sqrt{d}\varepsilon\log k) = \Omega(\log k)\text{OPT}_{\ell_2}(X).$$

$\qquad\square$

9 Fast Algorithm

In this section, we provide a fast variant of Algorithm 2 with running time $O(kd\log^2 k)$. The input of this algorithm is the set of reference centers $c^1, \ldots, c^k$ and the output is a threshold tree that splits all centers. The algorithm does not consider the data points (hence, it does not explicitly

assign them to clusters). It takes an extra $O(nk)$ time to assign every point in the data set to one of the leaves of the threshold tree.

This fast variant of Algorithm 2 picks a separate threshold cut ω^u for each leaf u. This cut is chosen uniformly at random from R^u, where

$$R^u = \bigcup_{c^i, c^j \in X_u} S_{ij}.$$

That is, R^u is the set of all cuts ω that separate at least two centers in X_u. The algorithm then splits leaf u into two parts using ω^u.

A straightforward implementation of the algorithm partitions each leaf by computing $\delta_c(\omega^u)$ for all centers c in X_u. It takes $O(d \cdot |X_u \cap C|)$ time to find R^u and sample ω^u for each u. It takes time $O(|X_u \cap C|)$ to split X_u into two groups. Thus, the total running time of this implementation of the algorithm is $O(k^2 d)$. We now discuss how to implement this algorithm with running time $O(kd \log^2 k)$ using red-black trees.

The improved algorithm stores centers for each leaf of the threshold tree in d red-black trees. Centers in the i-th red-black tree are sorted by the i-th coordinate. Using red-black trees, we can find the minimum and maximum values of c_i for $c \in C \cap X_u$ in time $O(d \log k)$. Denote these values by a_i and b_i, then

$$R^u = \bigcup_i \{i\} \times [a_i, b_i].$$

Hence, we can find R^u and sample a random cut ω^u in time $O(d \log k)$ for each u.

To partition set X^u into two groups with respect to $\omega^u = (i, \theta)$, we consider the i-th red-black tree for leaf u and find the sizes of the new parts, $Left = \{c \in X_u \cap C : c_i \leq \theta\}$ and $Right = \{c \in X_u \cap C : c_i > \theta\}$. We choose the set that contains fewer centers. Let us assume that the second set ($Right$) is smaller the first one ($Left$). Then, we find all centers in $Right$ and delete them from this red-black tree and all other red-black trees for node u. We assign the updated red-black trees (with deleted $Right$) to the left child of u. For the right child, we build d new red-black trees, which store centers for $Right$. Since we delete at most half of all centers in the red-black tree, each center is deleted at most $O(\log k)$ times. Each time it is deleted from d trees and inserted into d trees. Each deletion and insertion operation takes time $O(\log k)$. Thus, the total time of all deletion and insertion operations is $O(kd \log^2 k)$.

We note that though this algorithm slightly differs from the algorithm presented in Section 4, its approximation guarantees are the same.

References

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. Technical report, Stanford, 2006.

Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop. The hardness of approximation of euclidean k-means. *arXiv preprint arXiv:1502.03316*, 2015.

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.

Anup Bhattacharya, Dishant Goyal, and Ragesh Jaiswal. Hardness of approximation of euclidean k-median. *arXiv preprint arXiv:2011.04221*, 2020.

Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, 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.

L Breiman, JH Friedman, R Olshen, and CJ Stone. Classification and regression trees, 1984.

Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k-median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.

Moses Charikar and Lunjia Hu. Near-optimal explainable k-means for all dimensions. *arXiv preprint:2106.15566*, 2021.

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.

Sanjoy Dasgupta. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California, San Diego, 2008.

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.

Michael Elkin, Arnold Filtser, and Ofer Neiman. Terminal embeddings. *Theoretical Computer Science*, 697:1–36, 2017.

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. *arXiv preprint:2106.16147*, 2021.

Eduardo Laber and Lucas Murtinho. On the price of explainability for some clustering problems. *arXiv preprint arXiv:2101.01576*, 2021.

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. *SIAM Journal on Computing*, 45(2):530–547, 2016.

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.

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.

Nimrod Megiddo and Kenneth J Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.

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.