
Adapting k -means Algorithms for Outliers

Christoph Grunau^{*1} Václav Rozhoň^{*1}

Abstract

This paper shows how to adapt several simple and classical sampling-based algorithms for the k -means problem to the setting with outliers. Recently, Bhaskara et al. (NeurIPS 2019) showed how to adapt the classical k -means++ algorithm to the setting with outliers. However, their algorithm needs to output $O(\log(k) \cdot z)$ outliers, where z is the number of true outliers, to match the $O(\log k)$ -approximation guarantee of k -means++. In this paper, we build on their ideas and show how to adapt several sequential and distributed k -means algorithms to the setting with outliers, but with substantially stronger theoretical guarantees: our algorithms output $(1 + \varepsilon)z$ outliers while achieving an $O(1/\varepsilon)$ -approximation to the objective function. In the sequential world, we achieve this by adapting a recent algorithm of Lattanzi and Sohler (ICML 2019). In the distributed setting, we adapt a simple algorithm of Guha et al. (IEEE Trans. Know. and Data Engineering 2003) and the popular k -means|| of Bahmani et al. (PVLDB 2012). A theoretical application of our techniques is an algorithm with running time $\tilde{O}(nk^2/z)$ ¹ that achieves an $O(1)$ -approximation to the objective function while outputting $O(z)$ outliers, assuming $k \ll z \ll n$. This is complemented with a matching lower bound of $\Omega(nk^2/z)$ for this problem in the oracle model.

^{*}Equal contribution ¹ETH Zürich. Correspondence to: Christoph Grunau <cgrunau@ethz.ch>, Václav Rozhoň <rozhoňv@ethz.ch>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

¹ $\tilde{O}(\cdot)$ hides polylogarithmic factors in n . We assume that the minimum distance between any two points is 1 and the maximum distance between any two points Δ is bounded by $\text{poly}(n)$. Our algorithms work in general metric spaces and we assume a unit cost to evaluate the distance between two points.

1. Introduction

Clustering is a fundamental tool in machine learning and data analysis. It aims to partition a given set of objects into *clusters* in such a way that similar objects end up in the same cluster. The classical way of approaching the clustering problem is via the k -means formulation. In this formulation, one works with a set X consisting of n points in the d -dimensional Euclidean space \mathbb{R}^d and the objective is to output a set C consisting of k centers so as to minimize the sum of squared distances of points in X to C , i.e.,

$$\varphi(X, C) = \sum_{x \in X} \min_{c \in C} \|x - c\|^2.$$

Assigning each point to its closest cluster center naturally induces a partition of the points where nearby points tend to end up in the same partition.

k -means with Outliers One major drawback of k -means in practice is its sensitivity to outliers (GKL⁺17). This motivates optimizing a more robust version of the k -means objective. Probably the simplest such formulation is the *k -means with outliers* formulation (CKMN01). In this formulation, one additionally receives a number $z \in \{0, 1, \dots, n\}$. The aim is to find a set C consisting of k cluster centers and additionally a set $X_{out} \subseteq X$ consisting of z outliers so as to optimize the cost of *inliers* $X_{in} := X \setminus X_{out}$. That is, we optimize $\min_{X_{out}, C} \varphi(X \setminus X_{out}, C)$.

Known Results for k -means with Outliers It is known that the problem can be approximated up to a constant factor in polynomial time by rounding a certain linear program (Che08; KLS18). However, the complexity of these known algorithms is a large polynomial. Moreover, as we observe in Section 6, every constant factor approximation algorithm that outputs *exactly* z outliers needs to perform $\Omega(z^2)$ queries in the query model.

This motivates to weaken our requirements and look for fast algorithms that are allowed to output slightly more than z outliers. There is a line of work that makes progress toward such algorithms (CKMN01; MOP04; GKL⁺17; GLZ17; BVX19; IQM⁺20). However, to the best of our knowledge, known algorithms either need to output Az outliers, for some large constant $A \gg 1$ to obtain a reasonable approximation guarantee, or they suffer from at least a $\Omega(z^2)$

running time and, hence, to be truly applicable even for large z they need to be sped up by coresets constructions (FL11; HJLW18).

Our Contribution In this work we aim to further close the gap between theory and practice. We show how to adapt a number of classical sampling-based k -means algorithms to the setting with outliers. The main idea of the adaptations is a reduction to the variant of the problem with penalties described below. This is a known approach (CKMN01), but previous reductions of sampling based algorithms (BVX19) necessarily need to output Az outliers for some large constant $A \gg 1$, while we obtain algorithms that need to output only $(1 + \varepsilon)z$ outliers. More concretely, our contribution is threefold.

First, building on ideas of (LS19), we design a simple sampling-based algorithm with running time $\tilde{O}(nk/\varepsilon)$. It outputs an $O(1/\varepsilon)$ -approximate solution while declaring at most $(1 + \varepsilon)z$ points as outliers. This shows that sampling based algorithms that are known to be fast and have good practical performance can also achieve strong theoretical guarantees.

Second, we devise distributed algorithms for k -means with outliers where each machine needs to send only $\tilde{O}(k/\varepsilon)$ bits. Our construction achieves an $O(1)$ -approximation while outputting $(1 + \varepsilon)z$ outliers. Moreover, each machine only needs to perform polynomial-time computation. This improves on (LG18) who achieve an $(1 + \varepsilon)$ -approximation while outputting the same number of outliers as our algorithm, but their computation time is exponential.

Third, we show that one can achieve an $O(1)$ -approximation guarantee while discarding $O(z)$ outliers in time $\tilde{O}(k^2 \cdot n/z)$. This is done by speeding-up sampling with the Metropolis-Hastings algorithm as done in (BLHK16b; BLHK16a) together with additional ideas. This result is complemented by a matching lower bound of $\Omega(k^2 \cdot (n/z))$ for k -means/ k -median/ k -center algorithms that work for an arbitrary metric space accessed by distance queries. This improves on (MOP04; HJLW18) who give algorithms in this setting with a running time of $\tilde{O}(k^2 \cdot (n/z)^2)$. This is a significant improvement for $z \ll n$.

Roadmap. We overview the related work in Section 2 and our approach in Section 3. Then, we present our contributions in Sections 4 to 6 and our experiments in Section 7. Technical details are mostly deferred to appendices.

Notation We define $\varphi(x, C) = \min_{c \in C} \|x - c\|^2$ and set $\varphi(X, C) = \sum_{x \in X} \varphi(x, C)$. Similarly, we define $\tau_\Theta(x, C) = \min(\Theta, \varphi(x, C))$ and $\tau_\Theta(X, C) = \sum_{x \in X} \tau_\Theta(x, C)$. We call an algorithm an (α, β) -approximation if it outputs a set of k centers C and a set

of βz outliers X_{out} such that $\varphi(X \setminus X_{out}, C) \leq \alpha \text{OPT} = \alpha \varphi(X \setminus X_{out}^*, C^*)$, where OPT is the cost of a fixed optimal clustering C^* with a set of outliers $X_{out}^*, |X_{out}^*| = z$. Moreover, we define $X_{in} := X \setminus X_{out}$ and $X_{in}^* := X \setminus X_{out}^*$.

2. Previous Work

k -means Problem It is well-known that finding the optimal solution is NP-hard (ADHP09; MNV09). Currently the best known approximation ratio is roughly 6.36 (ANFSW19) and an approximation ratio of $(1 + \varepsilon)$ can be achieved for fixed dimension d (FRS19) or fixed k (KSS04). From the practical perspective, Lloyd’s heuristic (Llo82; BLSS16) is the algorithm of choice. As Lloyd’s algorithm converges only to a local optimum, it requires a careful seeding to achieve good performance. The most popular seeding choice is the k -means++ seeding (AV07; ORSS13). In k -means++ seeding (Algorithm 1) one chooses the first center uniformly at random from the set of input points X . In each of the following $k - 1$ steps, one samples a point in $x \in X$ as a new center with probability proportional to its current cost $\varphi(x, C)$. The seeding works well in practice and a theoretical analysis shows that even without running Lloyd’s algorithm subsequently it provides an expected $O(\log k)$ -approximation to the k -means objective (AV07).

Algorithm 1 k -means++ seeding

Input: X, k

- 1: Uniformly sample $c \in X$ and set $C = \{c\}$.
 - 2: **for** $i \leftarrow 2, 3, \dots, k$ **do**
 - 3: Sample $c \in X$ with probability $\varphi(c, C)/\varphi(X, C)$ and add it to C .
 - 4: **end for**
 - 5: **return** C
-

k -means with Outliers There is a growing body of research related to the k -means with outliers problem. On the practical side, Lloyd’s algorithm can be adapted to the noisy setting (CG13), but the output quality still remains dependent on the initial seeding. On the theoretical side, constant approximation algorithms based on the method of successive local search (Che08; KLS18) are known to provide a constant approximation guarantee. However, their running time is a large polynomial in n . We are interested in fast algorithms that are allowed to output slightly more than z outliers. Several algorithms have been proposed (CKMN01; MOP04; GKL⁺17; GLZ17; BVX19; IQM⁺20), but they either need at least $\Omega(z^2)$ time or they need to output at least Cz outliers for some $C \gg 1$. The algorithms can in general be sped up by coresets constructions (FL11; GKL⁺17; HJLW18; IQM⁺20). However, there is still a need for fast and simple algorithms with strong guar-

antees.

k -means with (Uniform) Penalties k -means with penalties is a different way of handling outliers introduced in the seminal paper of Charikar et al. (CKMN01). In the version of the problem with uniform penalties, we are given some positive number Θ and the goal is to output a set of centers $C = \{c_1, c_2, \dots, c_k\}$ so as to minimize the expression $\tau_\Theta(X, C) = \sum_{x \in X} \min(\Theta, \min_{c \in C} \|x - c\|^2)$. That is, the cost of any point is bounded by a threshold Θ .

It turns out that it is usually much simpler to work with k -means with penalties than k -means with outliers. This is quite helpful because results for k -means with penalties can be turned into results for k -means with outliers (CKMN01; LG18; BVX19; BR20). We also take this approach in this paper. We formalize the reduction in the next section and also present our improvement of it for sampling based algorithms.

3. Warmup: Reducing k -means with Outliers to k -means with Penalties

Our approach is based on reducing the problem of k -means with outliers to the problem of k -means with penalties. In this section, we first review how one can reduce the problem of k -means with outliers to the problem of k -means with penalties. Then we review an instance of this reduction by (BVX19) and provide a refined result which is the starting point of our work. We note that this can be seen as an instantiation of a general principle in optimization where one replaces a constraint with a penalty function known as Lagrangian relaxation (BBV04).

3.1. Review of the Previously Known Reduction

We start by giving the following lemma that formalizes how an α -approximate solution to the k -means with penalties objective can be used to obtain an algorithm providing an $(O(\alpha), O(\alpha))$ -approximate solution to the k -means with outliers objective.

Lemma 1 ((CKMN01)). *Let C be an α -approximate solution to the k -means with penalties objective with penalty $OPT/(2z) \leq \Theta \leq OPT/z$. Let X_{out} denote the set of points x for which $\tau_\Theta(x, C) = \Theta$. Then, it holds that $\varphi(X \setminus X_{out}, C) = O(\alpha OPT)$ and $|X_{out}| = O(\alpha z)$.*

Proof. Note that the optimal solution for k -means with outliers has a k -means with penalties cost upper bounded by $\varphi(X \setminus X_{out}^*, C^*) + |X_{out}^*| \Theta = OPT + z\Theta$. As C is an α -approximate solution to the k -means with penalties objective, we have $\varphi(X \setminus X_{out}, C) \leq \tau_\Theta(X, C) \leq \alpha(OPT + z\Theta) = O(\alpha OPT)$. Moreover, paying Θ for every $x \in X_{out}$ implies $|X_{out}| \leq \frac{\tau_\Theta(X, C)}{\Theta} \leq \frac{\alpha(OPT + z\Theta)}{\Theta} =$

$O(\alpha z)$. \square

We remark that the penalty Θ depends on OPT , so the algorithm for k -means with outliers needs to try this reduction for all $O(\log(n\Delta^2)) = O(\log n)$ powers of 2 between 1 and $n\Delta^2$. Here, $\Delta = n^{O(1)}$ is the maximum distance between any two points of X .

This reduction is very helpful as it allows us to easily adapt sampling-based algorithms like k -means++ to the setting with penalties since their analysis generalises to this setting. For the case of k -means++, this was shown in (BVX19; Li20) in the following theorem.

Algorithm 2 k -means++ (over)seeding with penalties

Input: A set of points X , k , ℓ , threshold Θ

- 1: Uniformly sample $c \in X$ and set $C = \{c\}$.
 - 2: **for** $i \leftarrow 2, 3, \dots, \ell$ **do**
 - 3: Sample $c \in X$ with probability $\tau_\Theta(c, C)/\tau_\Theta(X, C)$ and add it to C .
 - 4: **end for**
 - 5: **return** C
-

Theorem 1. [(AV07; BVX19; Li20)] *Suppose we run Algorithm 2 for $\ell = k$ steps. Then, the output set C is an $O(\log k)$ -approximation to the k -means with penalty Θ objective, in expectation.*

Proof sketch. The analysis of (AV07) proves this guarantee for Algorithm 1 for any $\varphi(a, b) := d^2(a, b)$ such that d is a metric. As the distance function $d'(a, b) = \min(d(a, b), \sqrt{\Theta})$ still defines a metric, one can thus directly use their analysis to prove Theorem 1. \square

More details are given in Appendix A. By Theorem 1, plugging in Algorithm 2 into Lemma 1 gives an $(O(\log k), O(\log k))$ -approximate algorithm for k -means++ with outliers. Moreover, running Algorithm 2 for $O(k)$ steps results in an $O(1)$ -approximation to the k -means objective in metric spaces (ADK09; BVX19). This gives the following tri-criteria approximation via Lemma 1: we get an $(O(1), O(1))$ -approximation algorithm that needs to use $O(k)$ centers.

3.2. Our Improved Reduction

The starting point of our work is the following improvement of the tri-criteria result from the previous subsection that enables us to get a constant factor approximation algorithm that outputs only $(1 + \varepsilon)z$ outliers, which is not possible by using Lemma 1 as a black box. The catch is that we need to use $O(k/\varepsilon)$ centers.

Theorem 2. Running Algorithm 2 for $\ell = O(k/\varepsilon)$ iterations and $OPT/(2\varepsilon z) \leq \Theta \leq OPT/(\varepsilon z)$ results in a set C with $\tau_\Theta(X_{in}^*, C) \leq 20OPT$, with positive constant probability.

Proof sketch (full proof in Appendix B). For an optimal set of centers $C^* = \{c_1^*, \dots, c_k^*\}$, we define $X_i^* \subseteq X_{in}^*$ as the subset of points $x \in X_{in}^*$ with $c_i^* = \arg \min_{c^* \in C^*} \varphi(x, c^*)$, where ties are broken arbitrarily. Fix one iteration of Algorithm 2 and let C be its current set of centers. We refer to a cluster X_i^* as *unsettled* if $\tau_\Theta(X_i^*, C) \geq 10\tau_\Theta(X_i^*, C^*)$.

Suppose that $\tau_\Theta(X_{in}^*, C) \geq 20OPT$, since otherwise we are already done. We sample a new point c from X_{in}^* with probability

$$\begin{aligned} \frac{\tau_\Theta(X_{in}^*, C)}{\tau_\Theta(X_{in}^*, C) + \tau_\Theta(X_{out}^*, C)} &\geq \frac{20OPT}{20OPT + \tau_\Theta(X_{out}^*, C)} \\ &\geq \frac{20OPT}{20OPT + OPT/\varepsilon} \geq \frac{\varepsilon}{2}, \end{aligned}$$

where we used $\tau_\Theta(X_{in}^*, C) \geq 20OPT$, $\tau_\Theta(X_{out}^*, C) \leq |X_{out}^*| \Theta \leq OPT/\varepsilon$, and that ε is small enough. Moreover, the cost of all settled clusters is bounded by $10OPT$, hence, given that c is sampled from X_{in}^* , the probability that it is sampled from an unsettled cluster is at least

$$\frac{\tau_\Theta(X_{in}^*, C) - 10OPT}{\tau_\Theta(X_{in}^*, C)} \geq \frac{20OPT - 10OPT}{20OPT} = \frac{1}{2},$$

where we again used $\tau_\Theta(X_{in}^*, C) \geq 20OPT$. Given that c is sampled from an unsettled cluster according to the τ_Θ -distribution, Corollary 1 in Appendix A tells us that the cluster becomes settled with probability at least $\frac{1}{5}$. Hence, we make a new cluster settled with probability at least $(\varepsilon/2) \cdot \frac{1}{2} \cdot \frac{1}{5} = \varepsilon/20$.

By a standard concentration argument, this implies that after $O(k/\varepsilon)$ steps of the algorithm, either all clusters are settled with positive constant probability, and hence we have $\tau_\Theta(X_{in}^*, C) \leq 10OPT$ and we are done, or during the course of the algorithm, the condition $\tau_\Theta(X_{in}^*, C) \geq 20OPT$ stopped being true, in which case we are again done. \square

Note that setting $\varepsilon = 1$ results in the same tri-criteria result we discussed in the previous subsection: This holds as $\tau_\Theta(X_{in}^*, C) = O(OPT)$ implies that $\tau_\Theta(X, C) \leq \tau_\Theta(X_{in}^*, C) + z\Theta = O(OPT/\varepsilon)$.

However, we can get more out of Theorem 2: note that as $\tau_\Theta(X_{in}^*, C) = O(OPT)$, for the set X_{out} defined as those $x \in X$ with $\tau_\Theta(x, C) = \Theta$, we have that $|X_{out} \cap X_{in}^*| = \frac{O(OPT)}{\Theta} = O(\varepsilon z)$. Hence, setting X_{out} as our output set of outliers gives on one hand

$$|X_{out}| \leq |X_{out}^*| + |X_{out} \cap X_{in}^*| = z + O(\varepsilon z).$$

On the other hand, we can bound

$$\begin{aligned} \varphi(X \setminus X_{out}, C) &= \tau_\Theta(X \setminus X_{out}, C) \\ &\leq \tau_\Theta(X, C) = O(OPT + z\Theta) = O(OPT/\varepsilon). \end{aligned}$$

Hence, we obtain an $O(1/\varepsilon)$ -approximation guarantee while outputting just $(1 + \varepsilon)z$ outliers.² By itself, Theorem 2 is still not satisfactory, as it requires us to oversample the number of centers by a factor of $O(1/\varepsilon)$. However, we next show three directions of improvement that lead to more interesting results.

4. Fast Sequential Algorithm

In this section, we present a simple sequential sampling-based algorithm for k -means with outliers that achieves an $O(1/\varepsilon)$ -approximation and outputs $(1 + \varepsilon)z$ outliers:

Theorem 3. For every $0 < \varepsilon < 1$, there exists an $(O(1/\varepsilon), 1 + \varepsilon)$ -approximation algorithm for k -means with outliers with running time $\tilde{O}(nk/\varepsilon)$.

The algorithm is based on ideas of a recent paper of Lattanzi and Sohler (LS19) who proposed augmenting k -means++ with $\tilde{O}(k)$ local search steps (KMN⁺04) as follows: their algorithm first invokes k -means++ to obtain an initial set of k centers. Afterwards, in each local search step, the algorithm samples a $(k + 1)$ -th point from the same distribution as k -means++. After sampling that point, the algorithm iterates over all $k + 1$ current centers and takes out the one whose deletion raises the cost the least (see Algorithm 3). Running k -means++ followed by $O(k)$ steps of Local-search++ is known to yield an $O(1)$ -approximation for the cost function $\varphi = \tau_\infty$ (CGPR20), with a positive constant probability.

Algorithm 3 One step of Local-search++

Input: X, C , threshold Θ

- 1: Sample $c \in X$ with probability $\tau_\Theta(c, C)/\tau_\Theta(X, C)$
 - 2: $c' \leftarrow \arg \min_{d \in C \cup \{c\}} \tau_\Theta(X, C \setminus \{d\} \cup \{c\})$
 - 3: **return** $C \setminus \{c'\} \cup \{c\}$
-

We get Theorem 3 by proving that $\tilde{O}(k/\varepsilon)$ iterations of Algorithm 3 with $OPT/(2\varepsilon z) \leq \Theta \leq OPT/(\varepsilon z)$ result in an $(O(1/\varepsilon), 1 + \varepsilon)$ -approximation guarantee. The analysis deals with new technical challenges and is deferred to Appendix C. We note that with minor changes to the original analysis of Lattanzi and Sohler, one can show that their result generalizes for arbitrary τ_Θ , similarly as Theorem 1, and, hence, by Lemma 1 one obtains an $(O(1), O(1))$ -approximation algorithm. Our refined analysis, crucially,

²In this particular case, we can even get an $O(1)$ -approximation guarantee by labelling the furthest $(1 + O(\varepsilon))z$ points as outliers, since the set $X_{out} \cup X_{out}^*$ has size at most $(1 + O(\varepsilon))z$ and $\varphi(X \setminus (X_{out} \cup X_{out}^*), C) \leq \tau_\Theta(X_{in}^*, C) = O(OPT)$.

does *not* use [Lemma 1](#) as a black box. Instead, we use the idea of the lemma as a building stone for the rather intricate analysis of the algorithm.

The intuition behind our proof of [Theorem 3](#) comes from [Theorem 2](#): the difference between running k -means++ for $O(k/\varepsilon)$ steps and the local search algorithm we described above is that the local search algorithm additionally removes one point after each sampling step. One can still hope that the increase in cost due to the removals is dominated by the decrease in cost due to the newly sampled center making an unsettled cluster settled, as we have seen in the analysis of [Theorem 2](#). This is indeed the case. We note that a local search based algorithm was also considered by ([GKL⁺17](#)), though with substantially weaker guarantees than our algorithm.

5. Distributed Algorithms for k -means with Outliers

To model the distributed setting, we consider the coordinator model where the input data $X = X_1 \sqcup X_2 \dots \sqcup X_m$ is split across m machines. Each machine can first perform some local computation. Then, each machine sends a message to a global coordinator who computes the final set of cluster centers C . The main complexity measure is the total number of bits each machine needs to send to the coordinator. An informal version of our main distributed result states the following.

Theorem 4. *There exists an $(O(1), 1 + \varepsilon)$ -approximate distributed algorithm in the coordinator model such that each machine sends at most $\tilde{O}(k/\varepsilon)$ many bits. Moreover, each machine only needs to perform polynomial-time computations.*

This improves on a recent result of ([LG18](#)). They give an algorithm with a $(1 + \varepsilon, 1 + \varepsilon)$ -approximation guarantee, but the required local running time is exponential. Other results similar to ours include ([GLZ17](#); [CAZ18](#)). Below, we sketch the high-level idea of the constructions and leave details to [Appendix F](#).

Simpler Construction In this paragraph we explain the high-level intuition behind a weaker result that provides an $(O(1/\varepsilon), 1 + \varepsilon)$ -approximation. This generalizes a classical construction of ([GMM⁺03](#)) that works in the k -means setting. For simplicity, we assume here that the machines know the value OPT and we define $\Theta := \text{OPT}/(\varepsilon z)$. This assumption can be lifted at the cost of a logarithmic overhead in the time - and message complexity.

Each machine starts by running [Algorithm 2](#) for $\tilde{O}(k/\varepsilon)$ steps with input X_j to obtain a set of centers Y_j . The sets of centers satisfy the property $\sum_j \tau_\Theta(X_j \cap X_{in}^*, Y_j) = O(\text{OPT})$. This follows from an argument very similar to

[Theorem 2](#) and will be important later on. Now, let $X_{out,j}$ denote all points in X_j that have a squared distance of at least Θ to the closest point in Y_j . All points in $X_{out,j}$ are declared as outliers and the remaining points in $X_j \setminus X_{out,j}$ are moved to the closest point in Y_j which creates a new, weighted instance X'_j . Each machine sends its weighted instance together with the number of declared outliers to the coordinator. The coordinator combines the weighted instances and finds an $(O(1/\varepsilon), 1 + \varepsilon)$ -approximate clustering using the algorithm guaranteed by [Theorem 9](#) on the instance X' , but with the number of outliers equal to $z' = z - |X_{out}| + O(\varepsilon z)$, where $X_{out} := \bigcup_j X_{out,j}$. Let C denote the corresponding set of cluster centers and X'_{out} denote the set consisting of the z' outliers. In [Theorem 15](#) in [Appendix F](#) we prove that $(C, X_{out} \cup X'_{out})$ is an $(O(1/\varepsilon), 1 + \varepsilon)$ -approximation.

The (simple) analysis follows from two observations. First, the number of inliers incorrectly labelled as outliers in the first step, i.e., $|\bigcup_j (X_{out,j} \cap X_{in}^*)|$, is bounded by $O(\varepsilon z)$. This follows from $\sum_j \tau_\Theta(X_j \cap X_{in}^*, Y_j) = O(\text{OPT})$ and $\tau_\Theta(x, Y_j) = \Theta$ for every $x \in X_{out,j}$. This ensures that in the end we output only $O(\varepsilon z)$ more outliers than if the coordinator ran [Algorithm 4](#) on the full dataset with original parameter z .

Second, the total movement cost of changing the instance X to instance X' is bounded by $\sum_j \varphi(X_j \setminus X_{out,j}, Y_j) \leq \sum_j \tau_\Theta(X_j, Y_j) \leq z\Theta + \sum_j \tau_\Theta(X_j \cap X_{in}^*, Y_j) = O(\text{OPT}/\varepsilon)$. Hence, the total cost changes additively by $O(\text{OPT}/\varepsilon)$ compared to the case where the coordinator runs [Algorithm 4](#) on the full dataset.

Refined Version To improve the approximation factor from $O(1/\varepsilon)$ down to $O(1)$ in [Theorem 4](#), we need to perform two changes. First, the coordinator runs the polynomial-time $(O(1), 1)$ -approximation algorithm of ([KLS18](#)) on the weighted instance instead of [Algorithm 4](#). Second, each machine records not only the number of points in $X_j \setminus X_{out,j}$ closest to each point $y \in Y_j$, but it additionally sends for each integer $k \in O(\log \Delta)$, how many of those points have a distance between 2^k and 2^{k+1} to y . The coordinator then constructs an instance based on this refined information.

k -means|| Adapting the popular k -means|| algorithm to the setting with outliers ([BMV⁺12](#)) can also be accomplished with a construction similar to the one explained above. The only difference is that instead of getting the weighted instance as a union of weighted instances from each machine, it is constructed in $O(\log n)$ sampling rounds by using the k -means|| idea: in each round we sample from the same distribution as in [Algorithm 2](#), but we sample $\tilde{O}(k/\varepsilon)$ points instead of just one point.

6. Tight Bounds for

$(O(1), O(1))$ -approximation Algorithms

In this section, we discuss tight bounds on the complexity of finding an $(O(1), O(1))$ -approximation for the k -means with outliers objective. In [Appendix D](#) we prove the following result.

Theorem 5. *There is an $(O(1), O(1))$ -approximation algorithm for k -means with outliers that runs in time $\tilde{O}(nk \cdot \min(1, k/z))$ and succeeds with positive constant probability.*

This result is based on [Theorem 2](#), but to speed up the algorithm from $\tilde{O}(nk)$ to $\tilde{O}(nk^2/z)$, we use the Metropolis-Hastings algorithm (with uniform proposal distribution) which was used in ([BLHK16b](#); [BLHK16a](#)) for k -means. Here, the idea is that in one sampling step of [Algorithm 2](#), instead of computing $\tau_\Theta(x, C)$ for all the points, we first subsample $\tilde{O}(n/z)$ points uniformly at random and only from those points we then sample one with probability roughly proportional to their current τ_Θ cost by defining a certain Markov chain. After this step, we end up with $O(k)$ cluster centers that match the guarantee of [Theorem 2](#), but the running time is improved to $\tilde{O}(nk^2/z)$. To reduce the number of cluster centers from $O(k)$ to k in sublinear time, we use additional sampling ideas to obtain a weighted instance that we can efficiently solve with our fast sequential algorithm. Independent of our work, ([DKP20](#)) devised an algorithm that obtains a set of $O(nk/z)$ cluster centers that match the guarantee of our $O(k)$ cluster centers. A slight variation of their algorithm can be implemented with running time $O(nk/z)$, faster than our $\tilde{O}(nk^2/z)$ runtime. However, it is not clear how to reduce the $O(nk/z)$ cluster centers to just k in sublinear time, as one cannot simply run our fast sequential algorithm on the subsampled set of cluster centers, but one additionally needs to define appropriate weights. Unlike in our case with $O(k)$ cluster centers, it is not clear how one can define appropriate weights for the $O(nk/z)$ cluster centers in time $\tilde{O}(nk^2/z)$.

Similarly, a result of ([HJLW18](#)) gives that running an $(O(1), O(1))$ -approximation algorithm on a uniform sample of $\tilde{O}(k(n/z)^2)$ points results in an $(O(1), O(1))$ -approximation without a need to compute clever weights for the subsampled points. However, this leads only to a suboptimal $\tilde{O}(k^2(n/z)^2)$ running time.

Lower Bound Next, we provide a matching lower bound. To that end, we restrict ourselves to the class of algorithms that work in an arbitrary metric space \mathcal{M} and access the distances of the space only by asking an *oracle* that upon getting queried on two points $x, y \in \mathcal{M}$ returns their distance.

Virtually all algorithms with guarantees we know of are of this type, possibly up to a constant loss in their approximation guarantee. In [Appendix A](#) we verify that this is the case also for algorithms that we consider here.

For the classical problems of k -means, k -median, and k -center, there is an $\Omega(nk)$ lower bound (i.e., showing that so many queries to the oracle are necessary) for metric space algorithms ([MP04](#); [Met02](#)). This essentially matches the complexity of k -means++ ([AV07](#)) or the `Local-search++` algorithm of Lattanzi and Sohler ([LS19](#)). The lower bound also holds in the setting with outliers if the output of the algorithm is an assignment of each of the n points to an optimal cluster, together with the list of the outliers. On the other hand, [Theorem 5](#) gives an $(O(1), O(1))$ algorithm with complexity $\tilde{O}(nk^2/z)$. The catch is that the output of this algorithm is just a set of centers and it does not compute for all the points their respective assignment to a closest center. Also, it does not label all the outliers. For this type of algorithms that only output the set of centers, we prove a matching lower bound of $\Omega(nk^2/z)$ in the metric space query model. The following theorem is proved in [Appendix E](#).

Theorem 6. *Any randomized algorithm for the k -means/ k -median/ k -center problem with outliers in the setting $k \geq C$, $z \geq Ck \log k$, and $n \geq Cz$ for an absolute constant C that with probability at least 0.5 gives an $(O(1), O(1))$ -approximation in the general metric space query model, needs $\Omega(nk^2/z)$ queries.*

Let us provide a brief intuition of the proof. The construction that yields [Theorem 6](#) is the following: we have $k/2$ large clusters and $k/2$ small clusters. All clusters are well separated and small clusters together contain $\Theta(z)$ points. As the algorithm can output only $\Theta(z)$ outliers, it needs to “find” $\Omega(k)$ small clusters. However, a point chosen from the input set uniformly at random is from a small cluster with probability $O(z/n)$ and we expect to need $\Omega(k)$ queries until we find out whether a point is from a small or a big cluster. This leads to the lower bound of $\Omega(k \cdot \frac{n}{z} \cdot k) = \Omega(nk^2/z)$ rounds.

Why it makes sense to consider bicriteria approximation Let us also observe a different lower bound against $(O(1), 1)$ -approximation algorithms that motivates why we are concerned with algorithms that can output $(1 + \varepsilon)z$ outliers. The following construction is, e.g., in ([Ind99](#)). Consider an input metric space with $z = n - 1$ and $k = 1$, where any two points have a distance of 1, up to a single pair x_1, x_2 whose distance is 0 (or some small ε). Any approximation algorithm (even a randomized one) that outputs exactly z outliers needs to “find” the pair x_1, x_2 and $\Omega(z^2)$ queries are needed for this. The example shows that there is a fundamental limit to the speed of $(O(1), 1)$ -approximation algorithms: for z linear in n they even need $\Omega(n^2)$ time. This should be

contrasted with the $(O(1), 1 + \varepsilon)$ -approximation algorithms that only need $\text{poly}(k/\varepsilon)$ time for $z = \Theta(n)$ that follows from (HJLW18).

7. Experiments

We tested the following algorithms on the datasets *kdd* (KDD Cup 1999) subsampled to 10 000 points with 38 dimensions and *spam* (Spambase) with 4601 points in 58 dimensions (DG17). We set the number of outliers z to be 10 percent of the dataset.

- **Lloyd**: Variant of Lloyd’s algorithm (Llo82; CG13) that handles outliers with random initialization (10 iterations);
- **k -means++**: k -means++ seeding (AV07);
- **k -means++ with penalties**: k -means++ with penalties (BVX19; Li20);
- **Metropolized k -means++ with penalties**: k -means++ with penalties, sped up by Metropolis-Hastings algorithm with 100 steps (see Appendix D);
- **Distributed k -means++ with penalties**: simplified variant of Algorithm 5 – input is partitioned in 10 subinputs, k -means++ with penalties is run on each of them with $\ell = 2k$, and weighted instances are sent to coordinator who runs k -means++ with penalties again to obtain the final k centers);
- **Sped up local search**: Variant of Algorithm 4 – k -means++ with penalties followed by k additional local search steps (for the objective with penalties).

To guess the value of Θ in all except the first two algorithms, we tried 10 values from 1 to 10^{10} , exponentially separated. The best solution was then picked and we followed by running 10 Lloyd iterations on it with the number of outliers for these iterations set to z (the same for the second k -means++ algorithm). The results for this setup for $k \in \{5, 10, \dots, 50\}$ are in Figs. 1 and 2, each value being an average over 10 runs.

k -means with penalties outperforms the first two baseline algorithms on average by around 40% in both datasets. Surprisingly, k -means++ seeding leads to consistently worse solutions than random initialization. We believe this indicates that the datasets indeed contain outliers that k -means++ picks preferably due to their large distance from other points. Distributed and metropolized variants are on par with k -means++ with penalties, except for the metropolized variant on the *spam* dataset. Sped up

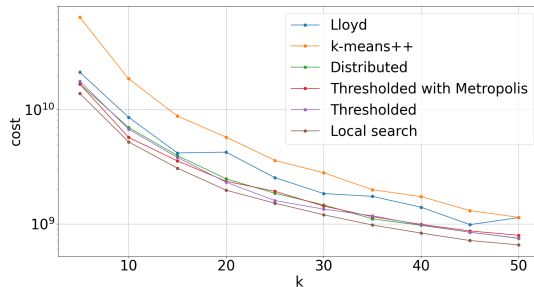


Figure 1. Experiments on kdd

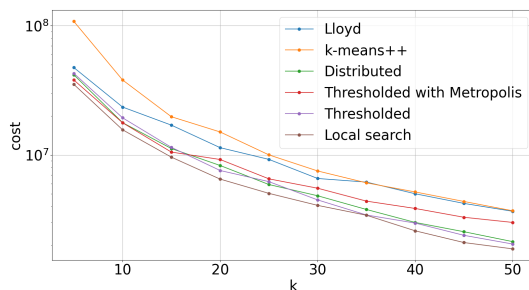


Figure 2. Experiments on spam

local search consistently outperforms k -means++ with penalties by around 12% in both datasets. It is also significantly slower, but we implemented only its simple $\tilde{O}(nk^2)$ implementation instead of the best possible $\tilde{O}(nk)$.

8. Conclusion

We have shown that several simple sampling-based algorithms for k -means can be adapted to handle outliers and retain strong theoretical guarantees, while still being similarly simple to implement. As a theoretical application, we showed that the complexity of getting an $(O(1), O(1))$ -approximation for k -means with outliers is $\tilde{\Theta}(nk^2/z)$ in the query model.

Acknowledgment

We thank Davin Choo, Mohsen Ghaffari, Saeed Ilchi, Andreas Krause, and Julian Portmann for engaging discussions. In particular, we thank Mohsen for his numerous helpful remarks and Davin and Saeed for sharing their code with us.

References

- [ADHP09] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.
- [ADK09] 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.
- [ANFSW19] 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*, pages FOCS17–97, 2019.
- [AV07] David Arthur and Sergei Vassilvitskii. k -means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [BBV04] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [BERS19] Anup Bhattacharya, Jan Eube, Heiko Röglin, and Melanie Schmidt. Noisy, greedy and not so greedy k -means++. *arXiv preprint arXiv:1912.00653*, 2019.
- [BLHK16a] Olivier Bachem, Mario Lucic, Hamed Hassani, and Andreas Krause. Fast and provably good seedings for k -means. In *Advances in neural information processing systems*, pages 55–63, 2016.
- [BLHK16b] Olivier Bachem, Mario Lucic, S Hamed Hassani, and Andreas Krause. Approximate k -means++ in sublinear time. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [BLSS16] Johannes Blömer, Christiane Lammersen, Melanie Schmidt, and Christian Sohler. Theoretical analysis of the k -means algorithm - A survey. *CoRR*, abs/1602.08254, 2016.
- [BMV⁺12] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k -means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.
- [BR20] Aditya Bhaskara and Aravinda Kanchana Rwanpathirana. Robust algorithms for online k -means clustering. In *Algorithmic Learning Theory*, pages 148–173, 2020.
- [BVX19] Aditya Bhaskara, Sharvaree Vadgama, and Hong Xu. Greedy sampling for approximate clustering in the presence of outliers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11148–11157. Curran Associates, Inc., 2019.
- [CAZ18] Jiecao Chen, Erfan Sadeqi Azer, and Qin Zhang. A practical algorithm for distributed clustering and outlier detection. In *Advances in neural information processing systems*, pages 2248–2256, 2018.
- [CG13] Sanjay Chawla and Aristides Gionis. k -means-: A unified approach to clustering and outlier detection. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 189–197. SIAM, 2013.
- [CGPR20] Davin Choo, Christoph Grunau, Julian Portmann, and Václav Rozhoň. k -means++: few more steps yield constant approximation, 2020.
- [Che08] Ke Chen. A constant factor approximation algorithm for k -median clustering with outliers. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 826–835, 2008.
- [CKMN01] Moses Charikar, Samir Khuller, David M Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 642–651. Society for Industrial and Applied Mathematics, 2001.
- [DG17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [DKP20] Amit Deshpande, Praneeth Kacham, and Rameshwar Pratap. Robust k -means++. In *Conference on Uncertainty in Artificial Intelligence*, pages 799–808. PMLR, 2020.
- [FL11] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.

- [FRS19] Zachary Friggstad, Mohsen Rezapour, and Mohammad R Salavatipour. Local search yields a ptas for k -means in doubling metrics. *SIAM Journal on Computing*, 48(2):452–480, 2019.
- [GKL⁺17] Shalmoli Gupta, Ravi Kumar, Kefu Lu, Benjamin Moseley, and Sergei Vassilvitskii. Local search methods for k -means with outliers. *Proceedings of the VLDB Endowment*, 10(7):757–768, 2017.
- [GLZ17] Sudipto Guha, Yi Li, and Qin Zhang. Distributed partial clustering. *CoRR*, abs/1703.01539, 2017.
- [GMM⁺03] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O’Callaghan. Clustering data streams: Theory and practice. *IEEE transactions on knowledge and data engineering*, 15(3):515–528, 2003.
- [HJLW18] Lingxiao Huang, Shaofeng Jiang, Jian Li, and Xuan Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 814–825. IEEE, 2018.
- [Ind99] Piotr Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 428–434, 1999.
- [IQM⁺20] Sungjin Im, Mahshid Montazer Qaem, Benjamin Moseley, Xiaorui Sun, and Rudy Zhou. Fast noise removal for k -means clustering, 2020.
- [KLS18] Ravishankar Krishnaswamy, Shi Li, and Sai Sandeep. Constant approximation for k -median and k -means with outliers via iterative rounding. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 646–659, New York, NY, USA, 2018. Association for Computing Machinery.
- [KMN⁺04] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. A local search approximation algorithm for k -means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.
- [KSS04] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1+\text{spl epsiv})$ -approximation algorithm for k -means clustering in any dimensions. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 454–462. IEEE, 2004.
- [LG18] Shi Li and Xiangyu Guo. Distributed k -clustering for data with heavy noise. *CoRR*, abs/1810.07852, 2018.
- [Li20] Min Li. The bi-criteria seeding algorithms for two variants of k -means problem. *Journal of Combinatorial Optimization*, 2020.
- [Liu96] Jun S Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and computing*, 6(2):113–119, 1996.
- [Llo82] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [LS19] Silvio Lattanzi and Christian Sohler. A better k -means++ algorithm via local search. In *International Conference on Machine Learning*, pages 3662–3671, 2019.
- [Met02] Ramgopal Reddy Mettu. Approximation algorithms for np -hard clustering problems, 2002.
- [MNV09] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k -means problem is np -hard. In *International Workshop on Algorithms and Computation*, pages 274–285. Springer, 2009.
- [MOP04] Adam Meyerson, Liadan O’callaghan, and Serge Plotkin. A k -median algorithm with running time independent of data size. *Machine Learning*, 56(1-3):61–87, 2004.
- [MP04] Ramgopal R Mettu and C Greg Plaxton. Optimal time bounds for approximate clustering. *Machine Learning*, 56(1-3):35–60, 2004.
- [MRR⁺53] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- [ORSS13] Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k -means problem. *Journal of the ACM (JACM)*, 59(6):1–22, 2013.

- [Roz20] Vaclav Rozhon. Simple and sharp analysis of k -means||, 2020.
- [Yao77] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 222–227. IEEE, 1977.

Appendices

A. Preparatory lemmas

More notation We start by setting up some more notation. Note that the optimal clustering C^* splits the dataset X into two parts: the set of *inliers* X_{in}^* with $\varphi(X_{in}^*, C^*) = \text{OPT}$ and the set of *outliers* X_{out}^* with $|X_{out}^*| = z$. Moreover, the set X_{in}^* is naturally split in k sets $X_1^*, X_2^*, \dots, X_k^*$, where X_j^* is the set of points $x \in X_{in}^*$ for which $\varphi(x, C) = \varphi(x, c_j^*)$. When using the notation $\tau_\Theta(X, C)$, we often drop the subscript Θ when it is clear from context and write just $\tau(X, C)$. For a single point c , we drop parentheses and write $\varphi(X, c)$ instead of $\varphi(X, \{c\})$ and similarly for τ . We define $\varphi^{-z}(X, C) = \min_{X_{out}, |X_{out}|=z} \varphi(X \setminus X_{out}, C)$. We use $\text{out}_\Theta(X, C)$ to denote the subset of points $x \in X$ such that $\tau_\Theta(x, C) = \Theta$ and $\text{in}_\Theta(X, C)$ to denote the subset of points $x \in X$ such that $\tau_\Theta(x, C) < \Theta$. We assume that the minimum distance between any two points in X is lower bounded by 1 and their maximum distance is upper bounded by Δ .

In this section we collect several useful lemmas that are either well-known or are variations of well-known results. First, we recall the well-known Chernoff bounds and then, for completeness, prove a generalization of sampling lemmas that were used to analyze k -means++ (AV07). These were, in slightly different forms, also proved in (Li20) or (BVX19).

Theorem 7 (Chernoff bounds). *Suppose X_1, \dots, X_n are independent random variables taking values in $\{0, 1\}$. Let X denote their sum. Then for any $0 \leq \delta \leq 1$ we have*

$$P(X \leq (1 - \delta)\mathbf{E}[X]) \leq e^{-\mathbf{E}[X]\delta^2/2}$$

and

$$P(X \geq (1 + \delta)\mathbf{E}[X]) \leq e^{-\mathbf{E}[X]\delta^2/3}.$$

Moreover, for any $\delta \geq 1$ we have

$$P(X \geq (1 + \delta)\mathbf{E}[X]) \leq e^{-\mathbf{E}[X]\delta/3}.$$

Next, we state basic facts about metric spaces and \mathbb{R}^d used to prove the sampling lemmas below.

Fact 1 (cf. (AV07)). *Let A be a subset of \mathbb{R}^d . Then*

$$\inf_{\mu \in \mathbb{R}^d} \varphi(A, \mu) = \varphi(A, \mu(A))$$

where $\mu(A) = \sum_{x \in A} x / |A|$ is the mean of A .

Fact 2 (cf. (AV07)). *Let $a, b, c \in M$ be three points in an arbitrary metric space. Then*

$$\varphi(a, c) \leq 2(\varphi(a, b) + \varphi(b, c))$$

and, more generally, for $C \subseteq M$ and $\varepsilon > 0$ we have

$$\varphi(a, C) - \varphi(b, C) \leq \varepsilon \cdot \varphi(b, C) + \left(1 + \frac{1}{\varepsilon}\right) \varphi(a, b),$$

where the former inequality is a special case for $\varepsilon = 1$ and $C = \{c\}$.

Proof. Let d denote the distance function of M . For $c \in C$ such that $d(b, C) = d(b, c)$ we have

$$\begin{aligned} \varphi(a, C) &= d^2(a, C) \leq d^2(a, c) \leq (d(a, b) + d(b, c))^2 \\ &= (d(a, b) + d(b, C))^2 = d^2(a, b) + d^2(b, C) + 2d(a, b)d(b, C) \\ &\leq d^2(a, b) + d^2(b, C) + \frac{1}{\varepsilon}d^2(a, b) + \varepsilon d^2(b, C) \\ &= \varphi(a, b) + \varphi(b, C) + \frac{1}{\varepsilon}\varphi(a, b) + \varepsilon\varphi(b, C) \end{aligned}$$

which is the the needed inequality, up to rearrangement. We used that

$$2d(a, b)d(b, C) \leq \frac{1}{\varepsilon}d^2(a, b) + \varepsilon d^2(b, C)$$

is equivalent to

$$(\sqrt{\varepsilon}d(b, C) - d(a, b)/\sqrt{\varepsilon})^2 \geq 0.$$

□

Fact 3. *Let M be an arbitrary metric space with distance function d . Then for any constant $T > 0$, the function $d' : d'(a, b) = \min(d(a, b), T)$ is also a distance function.*

Proof. For any $a, b, c \in M$, we need to prove $d'(a, c) \leq d'(a, b) + d'(b, c)$. If $d(a, b) \geq T$ or $d(b, c) \geq T$, respectively, we have $d'(a, b) = T$ or $d'(b, c) = T$, respectively, and, hence, $d'(a, b) + d'(b, c) \geq T \geq d'(a, c)$, as needed. Otherwise, we have $d'(a, c) \leq d(a, c) \leq d(a, b) + d(b, c) = d'(a, b) + d'(b, c)$, as needed. □

Fact 3 for example implies that in Fact 2 we can replace φ by τ_Θ , as $\tau_\Theta(a, b) = \min(d(a, b), \sqrt{\Theta})^2$. It also implies Theorem 1, as the analysis of (AV07) holds for an arbitrary metric space. However, this loses a factor of 2 in the approximation guarantee of Theorem 1 as we explain next. Proofs of Theorem 1 in (Li20; BVX19) instead generalize Lemmas 3.1 and 3.2 in (AV07) for τ costs to get Theorem 1; this can recover the same constants as in (AV07). For completeness, we also reprove these lemmas here as Lemmas 2 and 4 since we refer to them later in the proofs. Also, we prove a version of Lemma 2 for metric spaces as Lemma 3. This is the reason why arguments for arbitrary metric spaces lose additional factor of 2 in approximation guarantees. The relevance of Lemma 3 comes from the fact that it implies our algorithms work in arbitrary metric spaces, which is important for applications in Section 6.

Lemma 2 (cf. Lemma 3.1 in (AV07)). *For any cluster $A \subseteq \mathbb{R}^d$ we have*

$$\frac{1}{|A|} \sum_{c \in A} \tau_\Theta(A, c) \leq 2 \inf_{\mu \in \mathbb{R}^d} \tau_\Theta(A, \mu).$$

Proof. Recall that for the cost function φ we have $\frac{1}{|A|} \sum_{c \in A} \varphi(A, c) = 2\varphi(A, \mu(A))$ by Lemma 3.1 in (AV07). Let us fix an arbitrary $\mu \in \mathbb{R}^d$ and denote $A_{\text{in}} = \text{in}_\Theta(A, \mu)$ and $A_{\text{out}} = \text{out}_\Theta(A, \mu)$. Note that $\tau_\Theta(A, \mu(A_{\text{in}})) \leq \tau_\Theta(A, \mu)$ due to Fact 1. We have

$$\begin{aligned} & \frac{1}{|A|} \sum_{c \in A} \tau_\Theta(A, c) \\ &= \frac{1}{|A|} \left(\sum_{c \in A_{\text{in}}} \tau_\Theta(A_{\text{in}}, c) + \sum_{c \in A_{\text{out}}} \tau_\Theta(A_{\text{out}}, c) + \sum_{c \in A_{\text{out}}} \tau_\Theta(A, c) \right) \\ &\leq \frac{1}{|A|} \left(\sum_{c \in A_{\text{in}}} \varphi(A_{\text{in}}, c) + |A_{\text{in}}||A_{\text{out}}|\Theta + |A_{\text{out}}||A|\Theta \right) \\ &= \frac{1}{|A|} \left(2|A_{\text{in}}|\varphi(A_{\text{in}}, \mu(A_{\text{in}})) + |A_{\text{in}}||A_{\text{out}}|\Theta + |A_{\text{out}}||A|\Theta \right) \end{aligned}$$

Lemma 3.1 in (AV07)

$$\begin{aligned} &\leq 2\varphi(A_{\text{in}}, \mu(A_{\text{in}})) + 2|A_{\text{out}}|\Theta \\ &\leq 2\tau_\Theta(A, \mu). \end{aligned}$$

□

Lemma 3. *For any metric space M and subset of its points A we have*

$$\frac{1}{|A|} \sum_{c \in A} \varphi(A, c) \leq 4 \inf_{\mu \in M} \varphi(A, \mu).$$

Proof. For any $\mu \in M$, we can write

$$\begin{aligned}
 \frac{1}{|A|} \sum_{c \in A} \varphi(A, c) &= \frac{1}{|A|} \sum_{c \in A} \sum_{d \in A} \varphi(d, c) \\
 &\leq \frac{2}{|A|} \sum_{c \in A} \sum_{d \in A} \varphi(d, \mu) + \varphi(\mu, c) && \text{Fact 2} \\
 &= 4\varphi(A, \mu),
 \end{aligned}$$

as needed. □

The second sampling lemma states that sampling a point from a cluster A proportional to its τ -cost with respect to the current clustering C results in an 8-approximation of the cost of A .

Lemma 4 (cf. Lemma 3.2 in (AV07), Lemma 4 in (Li20) and Lemma 6 in (BVX19)). *For any cluster $A \subseteq \mathbb{R}^d$ and an arbitrary point set C , we have*

$$\sum_{c \in A} \frac{\tau_{\Theta}(c, C)}{\tau_{\Theta}(A, C)} \tau_{\Theta}(A, C \cup \{c\}) \leq 8 \inf_{\mu \in \mathbb{R}^d} \tau_{\Theta}(A, \mu).$$

Proof. Fix an arbitrary $\mu \in \mathbb{R}^d$ and $c, d \in A$. We have

$$\tau_{\Theta}(c, C) \leq 2(\tau_{\Theta}(c, d) + \tau_{\Theta}(d, C))$$

by Fact 2. For a given $c \in A$ we can then average over all $d \in A$ to get

$$\tau_{\Theta}(c, C) \leq \frac{2}{|A|} \sum_{d \in A} (\tau_{\Theta}(c, d) + \tau_{\Theta}(d, C)) = \frac{2}{|A|} (\tau_{\Theta}(A, c) + \tau_{\Theta}(A, C)). \quad (1)$$

This implies

$$\begin{aligned}
 &\sum_{c \in A} \frac{\tau_{\Theta}(c, C)}{\tau_{\Theta}(A, C)} \tau_{\Theta}(A, C \cup \{c\}) \\
 &\leq \sum_{c \in A} \frac{\frac{2}{|A|} (\tau_{\Theta}(A, c) + \tau_{\Theta}(A, C))}{\tau_{\Theta}(A, C)} \\
 &\sum_{d \in A} \min(\Theta, \varphi(d, C), \varphi(d, c)) && \text{Eq. (1)} \\
 &\leq \sum_{c \in A} \frac{\frac{2}{|A|} \tau_{\Theta}(A, c)}{\tau_{\Theta}(A, C)} \sum_{d \in A} \min(\Theta, \varphi(d, C)) \\
 &+ \sum_{c \in A} \frac{\frac{2}{|A|} \tau_{\Theta}(A, C)}{\tau_{\Theta}(A, C)} \sum_{d \in A} \min(\Theta, \varphi(d, c)) \\
 &= \sum_{c \in A} \frac{2}{|A|} \tau_{\Theta}(A, c) + \frac{2}{|A|} \sum_{c \in A} \tau_{\Theta}(A, c) \\
 &= \frac{4}{|A|} \sum_{c \in A} \tau_{\Theta}(A, c) \leq 8\tau_{\Theta}(A, \mu). && \text{Lemma 2}
 \end{aligned}$$

□

Both Lemma 2 and Lemma 4 were proven in the same way as corresponding lemmas in (AV07), and Theorem 1 follows from these lemmas in the same way as Lemma 3.3 in (AV07) follows from Lemmas 3.1 and 3.2 in (AV07) (cf. (BVX19; Li20)). Finally, we will use a simple corollary of Lemma 4.

Corollary 1. For any cluster $A \subseteq \mathbb{R}^d$ and an arbitrary point set C , sampling a point $c \in A$ proportional to $\tau_\Theta(c, C)$ results in

$$\tau_\Theta(A, C \cup \{c\}) \leq 10 \inf_{\mu \in \mathbb{R}^d} \tau_\Theta(A, \mu),$$

with probability at least $\frac{1}{5}$.

Proof. Follows from Lemma 4 by Markov inequality. \square

B. Basic Oversampling Result

In this section we prove a formal and more general version of Theorem 2. The full generality of the theorem is needed at a later point in time. To obtain Theorem 2, plug in some arbitrary $\Theta \in [\text{OPT}/(2\varepsilon z), \text{OPT}/(\varepsilon z)]$ and $\delta = 0.5$ in the following theorem.

Theorem 8 (cf. (ADK09)). Let $\delta, \varepsilon \in (0, 1]$ be arbitrary and suppose we run Algorithm 2 for $\ell = O(k/\varepsilon \cdot \log 1/\delta)$ steps to get output C . Then, with probability at least $1 - \delta$ we have

$$\tau_\Theta(X_{in}^*, C) = O(\text{OPT} + \varepsilon z \Theta).$$

Proof. We refer to a cluster X_i^* as unsettled if $\tau(X_i^*, C) \geq 10\tau(X_i^*, \mu(X_i^*))$. Fix one iteration of Algorithm 2 and let C be its current set of centers. Suppose that $\tau(X_{in}^*, C) \geq 20(\text{OPT} + \varepsilon z \Theta)$, since otherwise we are already done. We sample a new point c from X_{in}^* with probability

$$\frac{\tau(X_{in}^*, C)}{\tau(X_{in}^*, C) + \tau(X_{out}^*, C)} \geq \frac{20\varepsilon z \Theta}{20\varepsilon z \Theta + z \Theta} \geq \varepsilon/2,$$

where we used $\tau(X_{in}^*, C) \geq 20\varepsilon z \Theta$, $\tau(X_{out}^*, C) \leq |X_{out}^*| \Theta = z \Theta$, and $\varepsilon \leq 1$.

Moreover, given that c is sampled from X_{in}^* , the probability that it is sampled from an unsettled cluster is at least

$$\frac{\tau(X_{in}^*, C) - 10\text{OPT}}{\tau(X_{in}^*, C)} \geq \frac{20\text{OPT} - 10\text{OPT}}{20\text{OPT}} \geq \frac{1}{2},$$

where we used $\tau(X_{in}^*, C) \geq 20\text{OPT}$. Given that c is sampled from an unsettled cluster according to the τ -distribution, Corollary 1 tells us that the cluster becomes settled with probability at least $\frac{1}{5}$. Hence, we make a new cluster settled with probability at least $(\varepsilon/2) \cdot \frac{1}{2} \cdot \frac{1}{5} = \varepsilon/20$.

Finally, consider all $O(k/\varepsilon \cdot \log 1/\delta)$ iterations. We call each iteration *good*, if either $\tau(X_{in}^*, C) \leq 20(\text{OPT} + \varepsilon z \Theta)$, or we make a new cluster settled. As each iteration is good with probability at least $\varepsilon/20$, the expected number of good iterations is $\Omega(k \log 1/\delta)$. By Theorem 7, at least k iterations are good with probability at least $1 - e^{-\Omega(k \log 1/\delta)} \geq 1 - \delta$. This implies that in the end either $\tau(X_{in}^*, C) \leq 20(\text{OPT} + \varepsilon z \Theta)$ and we are done, or all clusters are settled. But this implies $\tau(X_{in}^*, C) \leq 10 \sum_i \tau(X_i^*, \mu(X_i^*)) \leq 10 \sum_i \varphi(X_i^*, \mu(X_i^*)) = 10\text{OPT}$, so we are again done. \square

C. Proof of Theorem 9

In this section we explain in more detail how to adapt the local-search algorithm of (LS19) in order to obtain an $(O(1/\varepsilon), 1 + \varepsilon)$ -approximation algorithm for k -means with outliers.

As explained in the main part of the paper, the main idea is to run the local-search algorithm with penalties. That is, each point is sampled as a new cluster center proportional to its τ_Θ -cost, where $\Theta = \Theta(\frac{\text{OPT}}{\varepsilon z})$. In total, we run $O(k \log \log k + k \log(1/\varepsilon)/\varepsilon)$ local-search steps. Hence, the algorithm can be implemented in time $\tilde{O}(nk/\varepsilon)$ (cf. (CGPR20), Section 3.3). After the first $O(k \log \log k)$ search steps, we show that the current cost is with constant probability at most $O(\text{OPT}/\varepsilon)$. The analysis is quite similar to the analysis of (LS19). Afterwards, within the next $O(k \log(1/\varepsilon)/\varepsilon)$ steps, we show that the number of points with a squared distance of at least 10Θ (The extra factor of 10 is needed in the analysis for technical reasons) drops below $(1 + \varepsilon)z$ with constant probability. Hence, we can declare all those points as outliers and the φ -cost of each remaining point is then only at most 10 times larger than its τ_Θ cost. Therefore, $\varphi(X, C)$ can still be bounded by $O(\text{OPT}/\varepsilon)$, as needed. The formal algorithm description is given below as Algorithm 4. Recall that Θ depends on OPT

and as OPT is unknown to the algorithm, it needs to be "guessed". Also note that due to technicalities in the analysis, the algorithm does not simply output the solution one obtains after running all the local-search steps, but it also considers all "intermediate" solutions as final solutions. The reason is that, unlike the total cost, the number of points with a squared distance of at least 10Θ is not necessarily decreasing monotonically.

Algorithm 4 Local-search++ with outliers

Input: $X, k, z, \varepsilon, \Delta$

Assumptions: $k + z < |X|, \varepsilon \in (0, 1]$

```

1:  $C \leftarrow \emptyset$ 
2:  $X_{out} \leftarrow \emptyset$ 
3: for  $i = 0$  to  $\lceil \log(n\Delta^2) \rceil$  do
4:    $OPT_{guess} \leftarrow 2^i, \beta \leftarrow 300/\varepsilon, \Theta \leftarrow \frac{\beta OPT_{guess}}{z}$ 
5:    $C_{i,0} \leftarrow k\text{-means++}(X, k, \Theta)$  (Algorithm 2)
6:   for  $j \leftarrow 1, 2, \dots, O(k \log \log k + k \cdot \frac{\log(1/\varepsilon)}{\varepsilon})$  do
7:      $C_{i,j} \leftarrow \text{Local-search++ step}(X, C_{i,j-1}, \Theta)$  (Algorithm 3)
8:      $X_{out}^{i,j} \leftarrow \text{out}_{10\Theta}(X, C)$ 
9:     if  $|X_{out}^{i,j}| \leq (1 + \varepsilon)z$  and  $\varphi(X \setminus X_{out}^{i,j}, C_{i,j}) < \varphi(X \setminus X_{out}, C)$  then
10:        $C \leftarrow C_{i,j}$ 
11:        $X_{out} \leftarrow X_{out}^{i,j}$ 
12:     end if
13:   end for
14: end for
15: Return  $(C, X_{out})$ 
    
```

Theorem 9 (Formal version of [Theorem 3](#)). *Let $\varepsilon \in (0, 1]$ be arbitrary and $X \subset \mathbb{R}^d$ be a set of n points. Furthermore, let $k \in \mathbb{N}$ and $z \in \mathbb{N}_0$ be such that $k + z < n$ (otherwise the cost is 0). Then, with positive constant probability, [Algorithm 4](#) returns a set X_{out} containing at most $(1 + \varepsilon)z$ points and a set of k cluster centers C such that*

$$\varphi(X \setminus X_{out}, C) = O(OPT/\varepsilon)$$

for $OPT = \min_{C', X'_{out}: |C'|=k, |X'_{out}|=z} \varphi(X \setminus X'_{out}, C')$.

Before going into the actual proof of [Theorem 9](#), we first need to introduce additional notation. As before, X denotes the set of all input points and X_{out}^* denotes the set of all input points that are declared as outliers by the optimal solution under consideration. Moreover, let $X_{far}^* \subseteq X \setminus X_{out}^*$ denote the set consisting of those points in $X \setminus X_{out}^*$ that have a squared distance of at least Θ to the closest optimal cluster center. Now, we define $X_{in}^* = X \setminus (X_{out}^* \cup X_{far}^*)$. In other words, we split $X = X_{in}^* \sqcup X_{far}^* \sqcup X_{out}^*$ (in [Section 2](#) we used $X = X_{in}^* \sqcup X_{out}^*$).

For a fixed candidate clustering C and for $i \in [k]$, $X_i \subseteq X_{in}^*$ denotes the set consisting of those points in X_{in}^* for which the closest center $c \in C$ is the center c_i . Similarly, let X_i^* denote the set consisting of those points in X_{in}^* that are clustered to the center c_i^* . Moreover, $X_{far,i}^*$ denotes the set of points in X_{far}^* that are closest to c_i among all candidate centers in C .

We say that the optimal cluster X_i^* is *ignored* by C if for every $x \in X_i^*$ we have $\tau_\Theta(x, C) = \Theta$. We define I to be the set of indices corresponding to ignored clusters. Next, we define a function $f: [k] \setminus I \mapsto [k]$ with $f(i) = \arg \min_{j \in [k]} d(c_i^*, c_j)$ for every $i \in [k] \setminus I$. Intuitively, the function f maps each optimal cluster that is not ignored to the closest candidate center (cf. [Fig. 3](#)). Now, for each candidate center, either zero, one or more than one optimal cluster maps to it. If none of the optimal clusters map to it, then we call the candidate center *lonely* and we denote the indices of all the lonely candidate centers by L . If exactly one optimal cluster maps to a given candidate center, then we call the candidate center *matched* and H denotes the set of indices of all the matched candidate centers. Without loss of generality, we assume that $f(i) = i$ for every $i \in H$. Finally, if more than one optimal cluster maps to a candidate center, then we call it *popular*. We refer to an optimal cluster X_i^* with cluster index $i \in [k] \setminus I$ as a *cheap* cluster if X_i^* maps to some popular candidate center with corresponding index j and, moreover, X_i^* has the smallest cost with respect to C among all optimal clusters that are mapped to the popular candidate center with index j . We let T denote the set of indices corresponding to cheap clusters. Let $b: [k] \setminus T \mapsto H \cup L$ be an arbitrary bijection with $b(i) = i$ for every $i \in H$ (cf. [Fig. 3](#)). That is, each optimal cluster center that is not cheap gets uniquely assigned to a candidate center which is not popular while preserving the matching between matched cluster centers.

Note that such a bijection exists since the number of cheap optimal clusters is equal to the number of popular candidate centers.

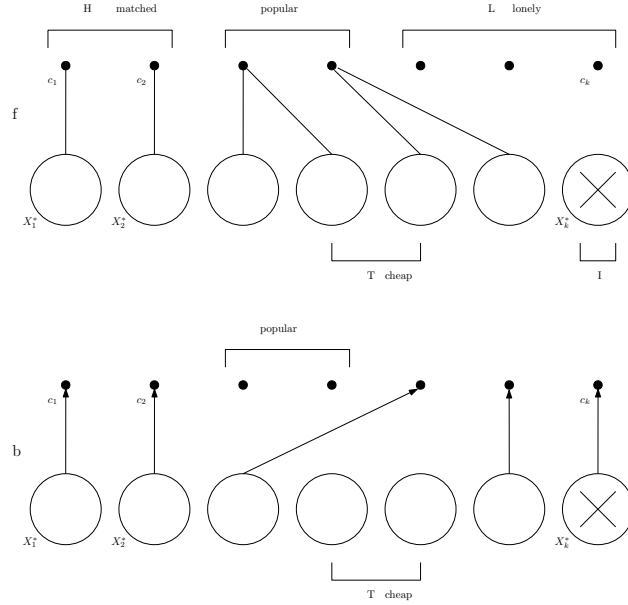


Figure 3. The figure captures several definitions. First, function f maps each optimal cluster X_i^* to its closest candidate center $c_j \in C$. Candidate centers with exactly one optimal cluster matched to them are denoted by H and for convenience we assume that f maps X_i^* to c_i there. Then there are popular centers with more than one match and for each such center we identify the cheapest matched cluster and put it in the set T . Finally, there are lonely centers with no matches. Ignored optimal clusters are not matched to any center by f . Second, we construct a bijection b that is used later for a double counting argument. In this bijection, we set $b(i) = f(i) = i$ for clusters i matched to centers in H . Then, we omit cheap clusters T and popular candidate centers, and extend b to an arbitrary bijection between the remaining clusters and lonely candidate centers.

For $i \in H \cup L$, we define

$$\text{outliers}(c_i) = \{x \in \text{in}_{10\Theta}(X_{out}^*, C) : i = \arg \min_{j \in [k]} \varphi(x, c_j)\}.$$

That is, $\text{outliers}(c_i)$ contains the real outliers that are closest to the candidate center c_i and which have a squared distance of at most 10Θ to c_i . The factor of 10 comes into play because of the following lemma.

Lemma 5. *Let C denote the current set of candidate centers and assume there exists some $x \in X_i^*$ with $\varphi(x, C) \geq 10\Theta$. Then, X_i^* is ignored by C .*

Proof. Let $x' \in X_i^*$ be arbitrary and let c' denote the candidate center in C that is closest to x' . We have

$$\begin{aligned} \|x' - c'\| &\geq \|x - c'\| - \|x' - x\| \\ &\geq \|x - c'\| - \|x' - c_i^*\| - \|x - c_i^*\| \\ &\geq \sqrt{10\Theta} - \sqrt{\Theta} - \sqrt{\Theta} \\ &> \sqrt{\Theta} \end{aligned} \quad x, x' \notin X_{far}^*$$

and therefore $\varphi(x', C) > \Theta$. Hence, X_i^* is ignored. \square

Now we define the notion of reassignment cost, similarly as it was done in (LS19). For each lonely candidate center c_ℓ , $\text{reassign}(X, C, c_\ell)$ is an upper bound for the increase in τ cost due to removing c_ℓ from the current set of candidate centers C . Similarly, for each matched candidate center c_h , $\text{reassign}(X, C, c_h)$ is an upper bound for the increase in cost due to removing c_h from the current set of candidate centers C , ignoring the cost increase for all points in the optimal cluster X_h^* .

Definition 1. Let $h \in H$ be arbitrary and let X_h^* be the optimal cluster that is captured by the center c_h . The reassignment cost of c_h is defined as

$$\begin{aligned} \text{reassign}(X, C, c_h) &= \tau(X_{in}^* \setminus X_h^*, C \setminus \{c_h\}) - \\ &\tau(X_{in}^* \setminus X_h^*, C) + \Theta|X_{far,h}| \\ &+ (\Theta|\text{outliers}(c_h)| - \tau(\text{outliers}(c_h), c_h)). \end{aligned}$$

For $\ell \in L$, the reassignment cost of c_ℓ is defined as

$$\begin{aligned} \text{reassign}(X, C, c_\ell) &= \tau(X_{in}^*, C \setminus \{c_\ell\}) - \tau(X_{in}^*, C) + \Theta|X_{far,\ell}| \\ &+ (\Theta|\text{outliers}(c_\ell)| - \tau(\text{outliers}(c_\ell), c_\ell)). \end{aligned}$$

Intuitively, the first two terms in the definition correspond to the increase of the cost for all the points in $X_{in}^* \setminus X_h^*$ and X_{in}^* , respectively. The third term corresponds to the increase of the cost for all points in X_{far}^* . This increase is bounded by Θ , their maximal cost. Finally, the last two terms in the definition below upper bound the increase of the cost of outliers. Here we again crudely upper bound the cost of each outlier after removing the candidate center by Θ .

Fact 4. For a matched cluster h we have

$$\text{reassign}(X, C, c_h) \geq \tau(X \setminus X_h^*, C \setminus \{c_h\}) - \tau(X \setminus X_h^*, C).$$

Similarly, for a lonely cluster ℓ we have

$$\text{reassign}(X, C, c_\ell) \geq \tau(X, C \setminus \{c_\ell\}) - \tau(X, C).$$

Proof. See above discussion. □

Lemma 6 (cf. Lemma 4 in (LS19)). For $i \in H \cup L$ we have

$$\begin{aligned} \text{reassign}(X, C, c_i) &\leq \frac{21}{100} \tau(\text{in}_\Theta(X_i, C), C) + 24\varphi(\text{in}_\Theta(X_i, C), C^*) + \Theta|X_{far,i}| \\ &+ \Theta|\text{outliers}(c_i)| - \tau(\text{outliers}(c_i), c_i). \end{aligned}$$

Proof. We only present the case $i \in H$, as the case $i \in L$ is similar and even easier. We observe that $\text{reassign}(X, C, c_i) = \tau(X_i \setminus X_i^*, C \setminus \{c_i\}) - \tau(X_i \setminus X_i^*, C) + \Theta|X_{far,i}| + \Theta|\text{outliers}(c_i)| - \tau(\text{outliers}(c_i), c_i)$, since vertices in clusters other than X_i will still be assigned to their current center. Hence, it suffices to show that

$$\begin{aligned} &\tau(X_i \setminus X_i^*, C \setminus \{c_i\}) - \tau(X_i \setminus X_i^*, C) \\ &\leq \frac{21}{100} \tau(\text{in}_\Theta(X_i, C), C) + 24\varphi(\text{in}_\Theta(X_i, C), C^*). \end{aligned}$$

As

$$\tau(\text{out}_\Theta(X_i \setminus X_i^*, C), C \setminus \{c_i\}) - \tau(\text{out}_\Theta(X_i \setminus X_i^*, C), C) = 0,$$

this is equivalent to showing that

$$\begin{aligned} & \tau(\text{in}_\Theta(X_i \setminus X_i^*, C), C \setminus \{c_i\}) - \tau(\text{in}_\Theta(X_i \setminus X_i^*, C), C) \\ & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_i, C), C) + 24\varphi(\text{in}_\Theta(X_i, C), C^*). \end{aligned}$$

The cost contribution of each point in $\text{in}_\Theta(X_i \setminus X_i^*, C)$ with respect to C is equal to the squared distance to the closest center in C and hence the analysis of Lattanzi and Sohler more or less directly applies. For the sake of completeness, we still present their analysis.

To upper bound the cost increase one incurs by removing c_i , we assign each point $p \in \text{in}_\Theta(X_i \setminus X_i^*, C) \cap X_j^*$, $j \neq i$, to the candidate center that captures the cluster X_j^* . By computing an upper bound for the clustering cost with respect to that assignment, we directly get an upper bound for $\tau(\text{in}_\Theta(X_i \setminus X_i^*, C), C \setminus \{c_i\})$. We upper bound the cost increase in two steps. First, we move each point $p \in \text{in}_\Theta(X_i \setminus X_i^*, C) \cap X_j^*$, $j \neq i$, to the cluster center c_j^* of X_j^* and we call the resulting multiset Q_i . Let q_p denote the point in Q_i corresponding to p . Among the centers in C , the closest center to q_p is now the center that captured the optimal cluster X_j^* . Thus, it is not equal to c_i , as c_i captures exactly one optimal cluster, namely X_i^* . Thus, we obtain

$$\begin{aligned} & \tau(q_p, C \setminus \{c_i\}) - \tau(p, C) \\ & = \tau(q_p, C) - \tau(p, C) && q_p \text{ not assigned to } c_i && (2) \\ & \leq \frac{1}{10} \tau(p, C) + 11\tau(p, q_p) && (\text{Facts 2 and 3 with } \varepsilon = 1/10) \\ & = \frac{1}{10} \tau(p, C) + 11\tau(p, C^*). \end{aligned}$$

This implies

$$\tau(q_p, C \setminus \{c_i\}) = \tau(q_p, C) \leq \frac{11}{10} \tau(p, C) + 11\tau(p, C^*). \quad (3)$$

Next, we upper bound the cost increase by moving each point $q_p \in Q_i$ back to its original position p . Formally, we bound

$$\begin{aligned} & \tau(p, C \setminus \{c_i\}) - \tau(q_p, C \setminus \{c_i\}) && (4) \\ & \leq \frac{1}{10} \tau(q_p, C \setminus \{c_i\}) + 11 \cdot \tau(p, q_p) && \text{Facts 2 and 3} \\ & \leq \frac{1}{10} \left(\frac{11}{10} \tau(p, C) + 11 \cdot \tau(p, C^*) \right) && (5) \\ & + 11 \cdot \tau(p, C^*) && \text{Eq. (3)} \\ & \leq \frac{11}{100} \tau(p, C) + 13 \cdot \tau(p, C^*) \end{aligned}$$

Combining the two inequalities Eqs. (2) and (4) yields

$$\begin{aligned} & \tau(p, C \setminus \{c_i\}) - \tau(p, C) && (6) \\ & = (\tau(p, C \setminus \{c_i\}) - \tau(q_p, C \setminus \{c_i\})) \\ & + (\tau(q_p, C \setminus \{c_i\}) - \tau(p, C)) \\ & \leq \frac{11}{100} \tau(p, C) + 13 \cdot \tau(p, C^*) + \frac{1}{10} \tau(p, C) + 11 \cdot \tau(p, C^*) \\ & \text{Eqs. (2) and (4)} \\ & \leq \frac{21}{100} \tau(p, C) + 24 \cdot \tau(p, C^*). \end{aligned}$$

Summing up the inequality Eq. (6) for all points $p \in \text{in}_\Theta(X_i \setminus X_i^*, C)$ then yields

$$\begin{aligned} & \tau(\text{in}_\Theta(X_i \setminus X_i^*, C), C \setminus \{c_i\}) - \tau(\text{in}_\Theta(X_i \setminus X_i^*, C), C) \\ & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_i \setminus X_i^*, C), C) + 24\tau(\text{in}_\Theta(X_i \setminus X_i^*, C), C^*) \\ & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_i, C), C) + 24\varphi(\text{in}_\Theta(X_i, C), C^*) \end{aligned}$$

and therefore

$$\begin{aligned} & \text{reassign}(X, C, c_i) \\ & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_i, C), C) + 24\varphi(\text{in}_\Theta(X_i, C), C^*) \\ & \quad + \Theta|X_{far,i}| + \Theta|\text{outliers}(c_i)| - \tau(\text{outliers}(c_i), c_i), \end{aligned}$$

as needed. □

C.1. Decreasing Cost to $O(1/\varepsilon)$

In this subsection, we will prove that while the cost of our solution is $\Omega(\text{OPT}/\varepsilon)$, one local search step decreases the cost by a factor of $1 - \Theta(1/k)$ with constant probability (Lemma 8). This means that after $\tilde{O}(k)$ steps, we have $\tau_\Theta(X, C) = O(\text{OPT}/\varepsilon)$ with constant probability (Lemma 9). This, by itself, is the easy part and our proof is essentially the analysis of Lattanzi and Sohler (LS19). The hard part is then the generalization to Lemma 11 in Section C.2, which, roughly speaking, shows that while we have $(1 + \varepsilon)z$ points with a big cost, we still do substantial progress in each step.

For the next definition, recall that b is the bijection between optimal clusters and candidate cluster centers that we defined earlier.

Definition 2. A cluster index $i \in [k] \setminus T$ is called awesome, if

$$\tau(X_i^*, C) - \text{reassign}(X, C, c_{b(i)}) - 9\tau(X_i^*, c_i^*) > \frac{\tau(X, C)}{100k}.$$

Intuitively, awesome clusters are such that sampling a point c from them leads to high improvement in cost if the sampled center is sampled close to the cluster mean. In particular, if $\tau(X_i^*, c) \leq 9\tau(X_i^*, c_i^*)$, swapping $c_{b(i)}$ with c leads to an improvement of $\tau(X, C)/(100k)$ in cost. The next lemma shows that we sample from an awesome cluster with constant probability.

Lemma 7. Let C denote the current set of candidate centers. Let $\varepsilon \in (0, 1]$ be arbitrary and $\beta := \frac{300}{\varepsilon}$. If $\tau(X, C) \geq 100\beta \text{OPT}$ and $\Theta \leq 2\frac{\beta \text{OPT}}{z}$, then

$$\sum_{i \in [k] \setminus T, i \text{ is awesome}} \tau(X_i^*, C) \geq \frac{1}{10} \tau(X, C).$$

Proof. We bound the cost of clusters that are not awesome.

$$\begin{aligned}
 & \sum_{i \in [k] \setminus T, i \text{ is not awesome}} \tau(X_i^*, C) \\
 & \leq \sum_{i \in [k] \setminus T} \text{reassign}(X, C, c_{b(i)}) \\
 & + 9\text{OPT} + \frac{\tau(X, C)}{100} \quad \text{Definition 2} \\
 & \leq \sum_{i \in [k] \setminus T} \left(\frac{21}{100} \tau(\text{in}_\Theta(X_{b(i)}, C), C) \right. \\
 & + 24\varphi(\text{in}_\Theta(X_{b(i)}, C), C^*) + \Theta |X_{\text{far}, b(i)}| \\
 & \left. + \Theta \cdot |\text{outliers}(c_{b(i)})| - \tau(\text{outliers}(c_{b(i)}), c_{b(i)}) \right) \quad \text{Lemma 6} \\
 & + 9\text{OPT} + \frac{\tau(X, C)}{100} \\
 & \leq \sum_{i \in H \cup L} \left(\frac{21}{100} \tau(\text{in}_\Theta(X_i, C), C) \right. \\
 & + 24\varphi(\text{in}_\Theta(X_i, C), C^*) + \Theta |X_{\text{far}, i}| \\
 & \left. + \Theta \cdot |\text{outliers}(c_i)| - \tau(\text{outliers}(c_i), c_i) \right) \\
 & + 9\text{OPT} + \frac{\tau(X, C)}{100} \quad b([k] \setminus T) = H \cup L \\
 & \leq \frac{21}{100} \tau(X, C) + 24\text{OPT} + \text{OPT} + z\Theta + \\
 & 9\text{OPT} + \frac{\tau(X, C)}{100} \quad \Theta |X_{\text{far}, i}| \leq \text{OPT}, X_i \subseteq X_{in}^* \\
 & \leq \frac{22}{100} \tau(X, C) + (2\beta + 34)\text{OPT} \quad \Theta \leq 2 \frac{\beta \text{OPT}}{z} \\
 & \leq \frac{30}{100} \tau(X, C). \quad \beta \geq 300, \tau(X, C) \geq 100\beta \text{OPT}
 \end{aligned}$$

Observe that from the definition of cheap clusters it follows that

$$\sum_{i \in T} \tau(X_i^*, C) \leq \frac{1}{2} \tau(X, C) \quad (7)$$

because every cheap cluster can be matched with a different cluster with at least as big cost that is also matched to the same popular cluster center.

Finally, we bound

$$\begin{aligned}
 & \sum_{i \in [k] \setminus T, i \text{ is awesome}} \tau(X_i^*, C) \\
 &= \tau(X_{in}^*, C) - \sum_{\substack{i \in [k] \setminus T, \\ i \text{ is not awesome}}} \tau(X_i^*, C) \\
 &\quad - \sum_{i \in T} \tau(X_i^*, C) \\
 &\geq \tau(X, C) - \tau(X_{out}^*, C) - \tau(X_{far}^*, C) \\
 &\quad - \frac{30}{100} \tau(X, C) - \frac{1}{2} \tau(X, C) \tag{Eq. (7)} \\
 &\geq \frac{20}{100} \tau(X, C) - z\Theta - \text{OPT} \\
 &\geq \frac{1}{10} \tau(X, C). \qquad \tau(X_{out}^*, C) \leq z\Theta, \tau(X_{far}^*, C) \leq \text{OPT} \\
 &\qquad \qquad \qquad \tau(X, C) \geq 100\beta \text{OPT}, \Theta \leq 2 \frac{\beta \text{OPT}}{z}
 \end{aligned}$$

□

Lemma 8. *Let C denote the current set of candidate centers. Let $\varepsilon \in (0, 1]$ be arbitrary and $\beta := \frac{300}{\varepsilon}$. Suppose that $\tau(X, C) \geq 100\beta \cdot \text{OPT}$ and $\Theta \leq 2 \frac{\beta \text{OPT}}{z}$. Then, with probability at least $1/100$, one local search step of [Algorithm 4](#) results in a new set of candidate centers C' with $\tau(X, C') \leq (1 - 1/(100k))\tau(X, C)$.*

Proof. Let c' denote the point sampled by `Local-search++` with outliers. The probability that $c' \in X_i^*$ for some i with i being an awesome cluster index is at least

$$\frac{(1/10)\tau(X, C)}{\tau(X, C)} = \frac{1}{10}$$

according to [Lemma 7](#). Let i denote an arbitrary awesome cluster index. [Lemma 4](#) implies that

$$\mathbf{E}[\tau(X_i^*, c') | c' \in X_i^*] \leq 8 \cdot \tau(X_i^*, c_i^*).$$

Hence, by a simple application of Markov's inequality, we obtain

$$\mathbf{P}[\tau(X_i^*, c') \leq 9 \cdot \tau(X_i^*, c_i^*) | c' \in X_i^*] \geq \frac{1}{9}.$$

Putting things together, the probability that the sampled point c' is contained in X_i^* for some awesome cluster index i and it additionally holds that

$$\tau(X_i^*, c') \leq 9 \cdot \tau(X_i^*, c_i^*) \tag{8}$$

is at least

$$\frac{1}{10} \cdot \frac{1}{9} > \frac{1}{100}.$$

Consider the case that $i \in H$. We can upper bound $\tau(X, C')$ as follows:

$$\begin{aligned}
 & \tau(X, C') \\
 & \leq \tau(X, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) \\
 & \text{swap } c_{b(i)} \text{ and } c' \\
 & = \tau(X, C) + (\tau(X, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) - \tau(X, C)) \\
 & = \tau(X, C) + \tau(X \setminus X_i^*, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) - \tau(X \setminus X_i^*, C) \\
 & \quad + \tau(X_i^*, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) - \tau(X_i^*, C) \\
 & \leq \tau(X, C) + \text{reassign}(X, C, c_i) + \tau(X_i^*, c') - \tau(X_i^*, C)
 \end{aligned}$$

Fact 4

$$\leq \tau(X, C) + \text{reassign}(X, C, c_i) + 9\tau(X_i^*, c_i^*) - \tau(X_i^*, C)$$

Eq. (8)

$$\leq \tau(X, C) - \frac{\tau(X, C)}{100k}$$

Definition 2

The case $i \notin H$ is very similar. Thus, we obtain that with probability at least $1/100$ we have

$$\tau(X, C') \leq (1 - 1/(100k)) \tau(X, C),$$

as desired. \square

Lemma 9. Let C denote the current set of candidate centers. Let $\varepsilon \in (0, 1]$ be arbitrary and $\beta := \frac{300}{\varepsilon}$. Let $\alpha > 0$ be arbitrary such that $\tau(X, C) \leq \alpha \cdot \beta \cdot \text{OPT}$. Furthermore, assume that $\Theta \leq 2 \frac{\beta \text{OPT}}{z}$. Let $C_0 := C$ and for $t \geq 0$, let C_{t+1} denote the set of centers obtained by running [Algorithm 4](#) with input centers C_t . Then, with positive constant probability $p > 0$, we have $\tau(X, C_T) \leq 100\beta \text{OPT}$ with $T := \lceil 1000 \cdot 100k \log \alpha \rceil = O(k \log(\alpha))$.

Proof. For $t \geq 0$, let X_t denote the indicator variable for the event that $\tau(X, C_{t+1}) > \max((1 - 1/(100k))\tau(X, C_t), 100\beta \text{OPT})$. According to [Lemma 8](#), $\mathbf{E}[X_t] \leq \frac{99}{100}$. Hence, with $X := \sum_{t=0}^{T-1} X_t$, we have $\mathbf{E}[X] \leq \frac{99}{100}T$. Thus, Markov's Inequality implies that there exists a constant $p > 0$ such that $\Pr[X \leq \frac{999}{1000}T] \geq p$. In that case, there are at least $100k \log \alpha$ iterations t for which $X_t = 0$ and therefore

$$\begin{aligned}
 & \tau(X, C_T) \\
 & \leq \max((1 - 1/(100k))^{100k \log \alpha} \tau(X, C_0), 100\beta \text{OPT}) \\
 & \leq \max(e^{-\log \alpha} \alpha \beta \text{OPT}, 100\beta \text{OPT}) \\
 & = 100\beta \text{OPT},
 \end{aligned}$$

as desired. \square

C.2. Decreasing Number of Outliers to $(1 + \varepsilon)z$

We are now ready to prove [Lemma 11](#). It roughly states that with probability $\Omega(\varepsilon)$ we sufficiently improve our solution, provided that there are at least $(1 + \varepsilon)z$ points with a squared distance of at least 10Θ to the current solution. [Lemma 12](#) then follows as a simple consequence of [Lemma 11](#). Roughly speaking, it states that starting with a set of centers C with a cost of $O((1/\varepsilon)\text{OPT})$, within $O(k/\varepsilon)$ local search steps, one obtains a set of centers C' such that at most $(1 + \varepsilon)z$ points have a squared distance of at least 10Θ to C' , with positive constant probability.

Let g be the number of points in X_{out}^* that have a squared distance of less than 10Θ to the closest center in C . Note that by

our definition of outliers(c), we have

$$\sum_{i \in [k]} |\text{outliers}(c_i)| = g \leq z. \quad (9)$$

Definition 3. A cluster index $i \in [k] \setminus T$ is called good, if

$$\tau(X_i^*, C) - \text{reassign}(X, C, c_{b(i)}) - 9\tau(X_i^*, c_i^*) > \frac{\tau(X, C) - z\Theta}{100k}.$$

Lemma 10. Let C denote the current set of candidate centers. Let $\varepsilon \in (0, 1]$ be arbitrary and $\beta := \frac{300}{\varepsilon}$. If for some $\varepsilon' \geq \varepsilon$, there are $(1 + \varepsilon')z$ different points with a squared distance of at least 10Θ to the closest candidate center in C and $\Theta \geq \frac{\beta \text{OPT}}{z}$, then

$$\sum_{i \in [k] \setminus T, i \text{ is good}} \tau(X_i^*, C) \geq \frac{\tau(X, C) - z\Theta}{10}.$$

Proof. Recall our assumption that exactly $(1 + \varepsilon')z$ points have a squared distance of at least 10Θ to the set C of candidate centers, i.e.,

$$|\text{out}_{10\Theta}(X, C)| = (1 + \varepsilon')z. \quad (10)$$

Moreover, we defined g as the number of points in X_{out}^* such that they have squared distance of less than 10Θ to C , so

$$|\text{out}_{10\Theta}(X_{out}^*, C)| = z - g. \quad (11)$$

Hence

$$|\text{out}_{10\Theta}(X_{in}^*, C)| \quad (12)$$

$$= |\text{out}_{10\Theta}(X, C)| - |\text{out}_{10\Theta}(X_{out}^*, C)| \quad (13)$$

$$- |\text{out}_{10\Theta}(X_{far}^*, C)|$$

$$\geq (1 + \varepsilon')z - (z - g) - |X_{far}^*|$$

Eqs. (10) and (11)

$$\geq \varepsilon'z + g - \frac{\text{OPT}}{\Theta}.$$

$$\text{OPT} \geq |X_{far}^*|\Theta$$

We will use the following bound

$$\begin{aligned} & \frac{\tau(X, C) - z\Theta}{100} \quad (14) \\ & \leq \frac{\tau(X_{in}^*, C) + \tau(X_{far}^*, C) + \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) - g\Theta}{100} \\ & \leq \frac{\tau(X_{in}^*, C) + \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) - g\Theta}{100} + \text{OPT}/100. \end{aligned}$$

Now we bound

$$\begin{aligned} & \sum_{i \in [k] \setminus T, i \text{ is not good}} \tau(X_i^*, C) \\ & \leq \left(\sum_{i \in [k] \setminus T} \text{reassign}(X, C, c_{b(i)}) \right) + 9\text{OPT} + \frac{\tau(X, C) - z\Theta}{100} \\ & \leq \sum_{i \in [k] \setminus T} \left(\frac{21}{100} \tau(\text{in}_{\Theta}(X_{b(i)}, C), C) + 24\varphi(\text{in}_{\Theta}(X_{b(i)}, C), C^*) \right) \end{aligned}$$

Definition 3

$$\begin{aligned}
 & + \Theta |X_{far, b(i)}| + \Theta \cdot |\text{outliers}(c_{b(i)})| \\
 & - \tau(\text{outliers}(c_{b(i)}), c_{b(i)}) + 9\text{OPT} + \frac{\tau(X, C) - z\Theta}{100}
 \end{aligned}$$

Lemma 6

$$\begin{aligned}
 & = \sum_{i \in H \cup L} \left(\frac{21}{100} \tau(\text{in}_\Theta(X_i, C), C) + 24\varphi(\text{in}_\Theta(X_i, C), C^*) \right. \\
 & + \Theta |X_{far, i}| + \Theta \cdot |\text{outliers}(c_i)| \\
 & \left. - \tau(\text{outliers}(c_i), c_i) \right) + 9\text{OPT} + \frac{\tau(X, C) - z\Theta}{100} \\
 & b([k] \setminus T) = H \cup L \\
 & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_{in}^*, C), C) + 24\text{OPT} + \text{OPT} \\
 & + \sum_{i \in H \cup L} (\Theta \cdot |\text{outliers}(c_i)| - \tau(\text{outliers}(c_i), c_i)) + 9\text{OPT} \\
 & + \frac{\tau(X, C) - z\Theta}{100} \\
 & X_i \subseteq X_{in}^* \\
 & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_{in}^*, C), C) + \sum_{i \in [k]} (\Theta \cdot |\text{outliers}(c_i)| - \tau(\text{outliers}(c_i), c_i)) \\
 & + 34\text{OPT} + \frac{\tau(X, C) - z\Theta}{100} \\
 & = \frac{21}{100} \tau(\text{in}_\Theta(X_{in}^*, C), C) + g\Theta - \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) \\
 & + 34\text{OPT} + \frac{\tau(X, C) - z\Theta}{100}
 \end{aligned}$$

Eq. (9)

$$\begin{aligned}
 & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_{in}^*, C), C) + g\Theta - \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) + 34\text{OPT} + \\
 & \frac{\tau(X_{in}^*, C) + \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) - g\Theta}{100} + \text{OPT}/100
 \end{aligned}$$

Eq. (14)

$$\begin{aligned}
 & \leq \frac{21}{100} \tau(\text{in}_\Theta(X_{in}^*, C), C) + \Theta g - \frac{1}{10} \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) \\
 & + 35\text{OPT} + \frac{\tau(X_{in}^*, C) - g\Theta}{100}
 \end{aligned}$$

Next, we make use of the inequality

$$\sum_{i \in T} \tau(X_i^*, C) \leq \frac{1}{2} \sum_{i \in [k], i \text{ is not ignored}} \tau(X_i^*, C). \quad (15)$$

This follows as a cheap cluster is not ignored, and each cheap cluster can be matched with a different cluster that is not ignored, has at least the same cost and is matched to the same popular cluster center.

Crucially, we now use Lemma 5 that says that whenever a cluster X_i^* is not ignored, we have $X_i^* = \text{in}_{10\Theta}(X_i^*, C)$.

Putting things together, we get

$$\sum_{i \in [k] \setminus T, i \text{ is good}} \tau(X_i^*, C)$$

$$\begin{aligned}
 &= \tau(X_{in}^*, C) - \sum_{i \in [k] \setminus T, i \text{ is not good}} \tau(X_i^*, C) - \sum_{i \in T} \tau(X_i^*, C) \\
 &\geq \tau(\text{in}_{10\Theta}(X_{in}^*, C), C) + \tau(\text{out}_{10\Theta}(X_{in}^*, C), C) \\
 &\quad - \left(\frac{21}{100} \tau(\text{in}_{10\Theta}(X_{in}^*, C), C) + \Theta g - \frac{1}{10} \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) \right) \\
 &\quad + 35\text{OPT} + \frac{\tau(X_{in}^*, C) - g\Theta}{100} \\
 &\quad - \frac{1}{2} \sum_{i \in [k], i \text{ is not ignored}} \tau(X_i^*, C) \tag{Eq. (15)} \\
 &\geq \frac{79}{100} \tau(\text{in}_{10\Theta}(X_{in}^*, C), C) + ((\varepsilon'z + g)\Theta - \text{OPT}) \\
 &\quad - \left(\Theta g - \frac{1}{10} \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) \right) \\
 &\quad + 35\text{OPT} + \frac{\tau(X_{in}^*, C) - g\Theta}{100} - \\
 &\quad \frac{1}{2} \tau(\text{in}_{10\Theta}(X_{in}^*, C), C) \tag{Lemma 5} \\
 &\geq \frac{1}{10} \tau(\text{in}_{10\Theta}(X_{in}^*, C), C) + \frac{1}{10} \tau(\text{in}_{10\Theta}(X_{out}^*, C), C) + \varepsilon'z\Theta \\
 &\quad - 100\text{OPT} - \frac{\tau(\text{out}_{10\Theta}(X_{in}^*, C), C) - g\Theta}{100} \\
 &\geq \frac{1}{10} \tau(\text{in}_{10\Theta}(X, C), C) + \varepsilon'z\Theta - 102\text{OPT} - \frac{\varepsilon'z\Theta}{100} \\
 &\tag{Eq. (12)} \\
 &\geq \frac{1}{10} (\tau(\text{in}_{10\Theta}(X, C), C) + \varepsilon'z\Theta) \tag{Eq. (10)} \\
 &\tag{Eq. (10)} \\
 &= \frac{1}{10} (\tau(\text{in}_{10\Theta}(X, C), C) + \tau(\text{out}_{10\Theta}(X, C), C) - z\Theta) \\
 &= \frac{\tau(X, C) - z\Theta}{10},
 \end{aligned}$$

as needed. \square

Lemma 11. *Let C denote the current set of candidate centers. Let $\varepsilon \in (0, 1]$ be arbitrary and $\beta := \frac{300}{\varepsilon}$. Suppose that for out current candidate centers C we have at least $(1 + \varepsilon)z$ points in X that have a squared distance of at least 10Θ to the closest center in C . Furthermore, assume that $\Theta \geq \frac{\beta \text{OPT}}{z}$. Then, with probability at least $\varepsilon/200$, one step of `Local-search++` with outliers results in a new set of candidate centers C' with $\tau(X, C') - z\Theta \leq (1 - 1/(100k))(\tau(X, C) - z\Theta)$.*

Proof. Let c' denote the point sampled by `Local-search++` with outliers. The probability that $c' \in X_i^*$ for some i with i being a good cluster index is at least

$$\begin{aligned}
 \frac{\frac{\tau(X, C) - z\Theta}{10}}{\tau(X, C)} &\geq \frac{(1 + \varepsilon)z\Theta - z\Theta}{10(1 + \varepsilon)z\Theta} & \tau(X, C) &\geq (1 + \varepsilon)z\Theta \\
 &\geq \frac{\varepsilon}{20}. & \varepsilon &\leq 1
 \end{aligned}$$

Let i denote an arbitrary good cluster index. [Lemma 4](#) implies that

$$\mathbf{E}[\tau(X_i^*, c') | c' \in X_i^*] \leq 8 \cdot \tau(X_i^*, c_i^*).$$

Hence, by a simple application of Markov's inequality, we obtain

$$\mathbb{P}[\tau(X_i^*, c') \leq 9 \cdot \tau(X_i^*, c_i^*) | c' \in X_i^*] \geq \frac{1}{9}.$$

Putting things together, the probability that the sampled point c' is contained in X_i^* for some good cluster index i and it additionally holds that

$$\tau(X_i^*, c') \leq 9 \cdot \tau(X_i^*, c_i^*) \quad (16)$$

is at least

$$\frac{\varepsilon}{20} \cdot \frac{1}{9} \geq \frac{\varepsilon}{200}.$$

First, consider the case that $i \in H$. In that case, we can upper bound $\tau(X, C')$ as follows:

$$\begin{aligned} \tau(X, C') &\leq \tau(X, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) && \text{swap } c_{b(i)} \text{ and } c' \\ &= \tau(X, C) + (\tau(X, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) - \tau(X, C)) \\ &= \tau(X, C) + \tau(X \setminus X_i^*, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) - \tau(X \setminus X_i^*, C) \\ &\quad + \tau(X_i^*, (C \setminus \{c_{b(i)}\}) \cup \{c'\}) - \tau(X_i^*, C) \\ &\leq \tau(X, C) + \text{reassign}(X, C, c_i) + \tau(X_i^*, c') - \tau(X_i^*, C) && \text{Fact 4} \\ &\leq \tau(X, C) + \text{reassign}(X, C, c_i) + 9\tau(X_i^*, c_i^*) - \tau(X_i^*, C) && \text{Eq. (16)} \\ &\leq \tau(X, C) - \frac{\tau(X, C) - z\Theta}{100k}. && \text{Definition 3} \end{aligned}$$

The case $i \notin H$ is very similar. Thus, we obtain

$$\begin{aligned} \tau(X, C') - z\Theta &\leq \left(\tau(X, C) - \frac{\tau(X, C) - z\Theta}{100k} \right) - z\Theta \\ &= (1 - 1/(100k)) (\tau(X, C) - z\Theta), \end{aligned}$$

as desired. □

Lemma 12. *Let C denote the current set of candidate centers. Let $\varepsilon \in (0, 1]$ be arbitrary and $\beta := \frac{300}{\varepsilon}$. Let $\alpha > 0$ be arbitrary such that $\tau(X, C) \leq \alpha \text{OPT}$. Assume that $\Theta \geq \frac{\beta \text{OPT}}{z}$. Let $C_0 := C$ and for $t \geq 0$, let C_{t+1} denote the set of centers obtained by running `Local-search++` with outliers with input centers C_t . Then, with positive constant probability $p > 0$, there are at most $(1 + \varepsilon)z$ points in X that have a squared distance of at least 10Θ to the closest candidate center in C_t for some $t \leq T$ with $T = O(\log(\alpha)k/\varepsilon)$.*

Proof. For $t \geq 0$, we define the potential $\Phi(t) = \tau(X, C_t) - z\Theta$. Now, let Y_t denote the indicator variable for the event that there are less than $(1 + \varepsilon)z$ points that have a squared distance of at least 10Θ to the closest center in C_t or $\Phi(t + 1) \leq (1 - 1/(100k))\Phi(t)$. Note that the random variables Y_0, Y_1, \dots, Y_{T-1} are not independent. However, Lemma 11 implies that

$$\mathbf{E}[Y_i | Y_0, Y_1, \dots, Y_{i-1}] \geq \frac{\varepsilon}{200}$$

for every $i \in \{0, \dots, T-1\}$ and every realization of random variables Y_0, Y_1, \dots, Y_{i-1} . Thus, the random variable $Y = \sum_{i=0}^{T-1} Y_i$ stochastically dominates the random variable $Y' = \sum_{i=0}^{T-1} Y'_i$, where the (Y'_i) 's are independent Bernoulli

variables that are equal to 1 with probability $\frac{\varepsilon}{200}$. Now, let $T' := \lceil 100k \cdot \log(\alpha) + 1 \rceil$ and $T := \lceil \frac{2T'}{\varepsilon/200} \rceil$. By a standard application of the Chernoff bound in [Theorem 7](#), we get

$$\mathbb{P}[Y' \leq T'] \leq P[Y' \leq (1 - 1/2)\mathbb{E}[Y']] \leq e^{-\Omega(\mathbb{E}[Y'])} = e^{-\Omega(1)}.$$

Thus, there exists a positive constant p such that with probability at least p , we have

$$\mathbb{P}[Y \geq T'] \geq p.$$

Assume that $Y \geq T'$. First, consider the case that there exists a $t \in \{0, \dots, T - 1\}$ for which there are less than $(1 + \varepsilon)z$ points in X that have a squared distance of at least 10Θ to the closest center in C_t . In that case we are done. Thus, assume that this does not happen. In particular, this implies that

$$\Phi(T - 1) \geq (1 + \varepsilon)z\Theta - z\Theta = \varepsilon\Theta z > \text{OPT}.$$

However, we also have

$$\begin{aligned} \Phi(T - 1) &\leq (1 - 1/(100k))^{T'-1} \Phi(0) \\ &\leq (1 - 1/(100k))^{100k \cdot \log(\alpha)} \alpha \text{OPT} \\ &\leq \text{OPT}, \end{aligned}$$

a contradiction. This concludes the proof. \square

C.3. Putting things together

Lemma 13. *Consider some iteration i for which $\text{OPT} \leq \text{OPT}_{\text{guess}} \leq 2\text{OPT}$. Then, with positive constant probability, there exists some iteration j of the inner loop such that $|X_{\text{out}}^{i,j}| \leq (1 + \varepsilon)z$ and $\varphi(X \setminus X_{\text{out}}^{i,j}, C_{i,j}) = O(1/\varepsilon)\text{OPT}$.*

Proof. According to ([BVX19](#); [Li20](#)), running [Algorithm 2](#) results in a set of centers $C_{i,0}$ such that

$$\begin{aligned} \mathbb{E}[\tau(X, C_{i,0})] &= O(\log k) \cdot \min_{C': |C'|=k} \tau(X, C') \\ &\leq O(\log k) (\text{OPT} + z \cdot \Theta) \leq O(\log k)\beta\text{OPT}. \end{aligned}$$

By using Markov's inequality, we therefore have $\tau(X, C_{i,0}) = O(\log k)\beta\text{OPT}$ with positive constant probability. Conditioned on that event, we use [Lemma 9](#) to deduce that $\tau(X, C_{i,T}) \leq 100\beta\text{OPT}$ for some $T = O(k \log(O(\log k))) = O(k \log \log k)$ with positive constant probability. Then, we use [Lemma 12](#) to deduce that with positive constant probability, there exists some $T' \in O(\log(1/\varepsilon)k/\varepsilon)$ such that there are at most $(1 + \varepsilon)z$ points in X with a squared distance of at least 10Θ to the closest center in $C_{i,T+T'}$ and furthermore

$$\tau(X, C_{i,T+T'}) \leq \tau(X, C_{i,T}) \leq 100\beta\text{OPT} = O(1/\varepsilon)\text{OPT}.$$

Hence, $|X_{\text{out}}^{i,T+T'}| \leq (1 + \varepsilon)z$ and

$$\varphi(X \setminus X_{\text{out}}^{i,T+T'}, C_{i,T+T'}) \leq 10 \cdot \tau(X \setminus X_{\text{out}}^{i,T+T'}, C_{i,T+T'}) \leq O(1/\varepsilon)\text{OPT}$$

with $T + T' = O(k \log \log k + \log(1/\varepsilon)k/\varepsilon)$, as set in the inner loop of [Algorithm 4](#). \square

Finally, we are ready to finish the analysis of [Algorithm 4](#) by proving [Theorem 9](#).

Proof of Theorem 9. As $k + z < n$, we have

$$1 \leq \min_{C', X'_{out}: |C'|=k, |X'_{out}|=z} \varphi(X \setminus X'_{out}, C) \leq n\Delta^2.$$

For $i = 0$, we have $2^i = 1$ and for $i = \lceil \log(n\Delta^2) \rceil$, we have

$$2^i = 2^{\lceil \log(n\Delta^2) \rceil} \geq n\Delta^2.$$

Thus, there exists some $i \in \{0, 1, \dots, \lceil \log(n\Delta^2) \rceil\}$ such that in the i -th iteration we have $OPT_{guess} \in [\text{OPT}, 2\text{OPT}]$ and the statement follows from Lemma 13. \square

D. Using the Metropolis-Hastings algorithm to speed up Algorithm 2

Here we explain how to speed up Algorithm 2 by using the Metropolis-Hastings algorithm. This was first done in (BLHK16b; BLHK16a) for the classical k -means++ algorithm, but there it only leads to a rather weak additive error guarantee, which turns out not to be the case for k -means with outliers.

We alter every step of Algorithm 2 as follows. Instead of sampling a new point proportional to its τ_Θ -weight, we instead sample a new point from the following Markov chain: First, we sample a point x according to a proposal distribution q . Then, in the following T steps, we always sample a new point y from q and set $x \leftarrow y$ with probability $\min\left(1, \frac{q(x)\pi(y)}{q(y)\pi(x)}\right)$, where π is the target distribution, so in our case, the $\tau_\Theta(\cdot, C)/\tau_\Theta(X, C)$ -distribution. In other words, we define a Markov chain with transition probability

$$P(x, y) = q(y) \cdot \min\left(1, \frac{q(x)\pi(y)}{q(y)\pi(x)}\right).$$

In (BLHK16b), a uniform proposal distribution $q(x) = 1/n$ for all $x \in X$ was chosen. The advantage of the uniform distribution is that one can sample from it easily, but only arguably rather weak guarantees are known for the resulting algorithm in the context of speeding up k -means++. In (BLHK16a), $q(x) = \varphi(x, c)/\varphi(X, c)$ for a random point c was chosen. This distribution can be precomputed in $O(n)$ time and the guarantee of the resulting algorithm matches the guarantee of the original k -means++ algorithm up to an additional additive error term $\delta\varphi(X, \mu(X))$, if $T = \tilde{O}(1/\delta)$ is chosen.

Remark 1. *One can see (but we will not prove it here) that the analysis of (BLHK16a) directly generalizes to τ_Θ -costs for each non-negative Θ . Choosing $\Theta = \text{OPT}/z$, one obtains $\delta\tau_\Theta(X, \mu(X)) \leq \delta \cdot n\Theta = \frac{\delta n \text{OPT}}{z}$. Hence, the choice of $\delta = O(z/n)$ yields that the additional incurred error is of the same order as OPT . Hence, generalizing the result of (BLHK16a) and using Lemma 1, we obtain an $(O(\log k), O(\log k))$ -approximation algorithm with time complexity of $\tilde{O}(n + k \cdot (k/\delta)) = \tilde{O}(n + nk^2/z)$, because we need to precompute the proposal distribution q in $O(n)$ time and sampling one point takes $\tilde{O}(1/\delta)$ steps, each implementable in time $O(k)$. Changing proposal distribution to uniform would recover the same guarantees.*

While in the case of classical k -means, one needs to use the more complicated proposal distribution $q(x) = \varphi(x, c)/\varphi(X, c)$ to get some guarantees, it turns out that for k -means with outliers, uniform proposal suffices. To get Theorem 5, we start by proving that Algorithm 2 with $\ell = O(k)$ can be sped up with Metropolis-Hastings algorithm with uniform proposal.

Theorem 10. *A version of Algorithm 2 with $\ell = O(k)$, $\Theta = \Theta(\text{OPT}/z)$ and where each sampling step is approximated by the Metropolis-Hastings algorithm running for $T = O(n/z)$ steps and a uniform proposal distribution will run in time $\tilde{O}(\frac{nk^2}{z})$ and produce a solution that with positive constant probability satisfies*

$$\tau_\Theta(X_{in}^*, C) = O(\text{OPT}).$$

To prove Theorem 10, we first recall a classical result about the Metropolis-Hastings algorithm (MRR⁺53; Liu96). If the proposal distribution q and the target distribution π satisfy $q(x) \geq \delta\pi(x)$ for all x and some $\delta > 0$, then after $T = O(1/\delta)$ steps, the distribution of the Metropolis-Hastings chain dominates $\pi/2$.

Lemma 14 (one Metropolis-Hastings step (Liu96)). *Let π denote the target distribution we want to approximate with Metropolis-Hastings and $p_t = \alpha\pi + (1 - \alpha)p'_t$ for some distribution p'_t . Let q be a proposal distribution with $q(x) \geq \delta\pi(x)$ for all x . Then, after one step of Metropolis-Hastings, we are in a distribution p_{t+1} which can be written as $p_{t+1} = \alpha'\pi + (1 - \alpha')p'_{t+1}$ for $\alpha' = \alpha + \delta(1 - \alpha)$.*

Proof. For any fixed vertex x , given that x is the current vertex after t steps, the probability of moving from x to y is

$$P(x, y) = q(y) \cdot \min\left(1, \frac{q(x)\pi(y)}{q(y)\pi(x)}\right).$$

Now, if $q(x)\pi(y) \geq q(y)\pi(x)$, we have

$$P(x, y) = q(y) \geq \delta\pi(y)$$

by the definition of δ . On the other hand, if $q(x)\pi(y) < q(y)\pi(x)$, we have

$$P(x, y) = \frac{q(x)\pi(y)}{\pi(x)} \geq \frac{q(x)\pi(y)}{q(x)/\delta} = \delta\pi(y).$$

So, for any x and any y we have $P(x, y) \geq \delta\pi(y)$, and since π is a stationary distribution, for any x we have $p_{t+1}(x) \geq \alpha\pi(x) + (1 - \alpha)\delta\pi(x)$, as needed. \square

We now prove [Theorem 10](#).

Proof. Fix one sampling step of [Algorithm 2](#) and suppose that $\tau(X_{in}^*, C) \geq 40\text{OPT}$. For $T = O(n/z)$, we will show that we sample from a distribution p_T such that for $\pi(x) = \tau(x, C)/\tau(X, C)$ we have $p_T(x) \geq \pi(x)/2$ for all x . Hence, we may interpret sampling from p_T as sampling from π with probability $1/2$ and sampling from some other distribution otherwise. Thus, we have essentially reduced the analysis to the one in the proof of [Theorem 8](#), only loosing a 2-factor in the number of required iterations.

The running time of the algorithm is $O(k \cdot \frac{n}{z} \cdot k)$, as we need to sample $O(k)$ points in total, $O(n/z)$ iterations of the Metropolis-Hastings algorithm are necessary for each point sample and in each step we need to compute $\tau_\Theta(x, C)$, which can be done in time proportional to the number of points already sampled.

We now prove that $p_T(x) \geq \pi(x)/2$ for all x , provided that $\Theta \in [\text{OPT}/z, 2\text{OPT}/z]$. First note that $\pi(x) = \tau(x, C)/\tau(X, C) \leq \Theta/\tau(X, C)$, so for $\delta = \frac{z}{n}$ we have for any x that

$$\delta\pi(x) \leq \frac{z}{n} \frac{2\text{OPT}/z}{\tau(X, C)} = q(x) \frac{2\text{OPT}}{\tau(X, C)} \leq q(x).$$

Hence, we may use [Lemma 14](#) to conclude that after $O(1/\delta) = O(n/z)$ steps, $p_T(x) \geq \pi(x)/2$ for all x , as needed. \square

We are now ready to prove [Theorem 5](#) that we restate here for convenience.

Theorem 5. *There is an $(O(1), O(1))$ -approximation algorithm for k -means with outliers that runs in time $\tilde{O}(nk \cdot \min(1, k/z))$ and succeeds with positive constant probability.*

Proof. If $z = O(k \log k)$ or $z = O(\log^2 n)$, we can run [Algorithm 4](#) with $\varepsilon = 1$ to obtain an $(O(1), O(1))$ -approximation in time $\tilde{O}(nk)$ ([Theorem 9](#)). Otherwise, we run the Metropolis-Hastings based variant of [Algorithm 2](#) from [Theorem 10](#) with $\ell = O(k)$ that runs in $\tilde{O}(nk^2/z)$ time for $O(\log n)$ values $\Theta_0 = 2^0/z, \Theta_1 = 2^1/z, \dots, \Theta_{\log(\Delta^2 n)} = \Delta^2 n/z$. Hence, we get a sequence of sets of centers Y_0, Y_1, \dots with $|Y_i| = O(k)$. [Theorem 10](#) ensures that there will exist one value $\tilde{\Theta}$ with $\text{OPT}/(2z) \leq \tilde{\Theta} \leq \text{OPT}/z$ in the set of values $\Theta_0, \dots, \Theta_{\log(\Delta^2 n)}$ such that the sped up [Algorithm 2](#) returns, with positive constant probability, a set Y with $\tau_{\tilde{\Theta}}(X_{in}^*, Y) = O(\text{OPT})$.

In [Theorem 11](#) we give an algorithm with running time $\tilde{O}(nk^2/z)$ that turns the pair $(\tilde{Y}, \tilde{\Theta})$ into a set of cluster centers $\tilde{C}, |\tilde{C}| = k$, such that $\varphi^{-Az}(X, \tilde{C}) = O(\text{OPT})$ for some universal constant A , with positive constant probability.

We apply the algorithm from [Theorem 11](#) to all pairs (Y_i, Θ_i) and get a sequence of $O(\log n)$ sets $\tilde{C}_0, \tilde{C}_1, \dots$ with $|\tilde{C}_i| = k$ such that, by [Theorem 11](#), with positive constant probability it contains a set \tilde{C} with $\varphi^{-Az}(X, \tilde{C}) = O(\text{OPT})$.

To conclude, we describe how in time $\tilde{O}(nk/z)$ we can get an estimator ζ_C of $\varphi^{-Az}(X, C)$ for some set of k centers C such that the following two properties are satisfied.

$$\zeta_C \leq 4\varphi^{-Az}(X, C) \quad \text{with probability } 1 - 1/\text{poly}(n) \quad (17)$$

and

$$\zeta_C \geq \frac{1}{4}\varphi^{-10Az}(X, C) \quad \text{with probability } 1 - 1/\text{poly}(n). \quad (18)$$

Then, we can simply estimate the cost of each set of centers C_i and return the one with the smallest estimated cost. With positive constant probability, there is at least one set of centers C_i with $\varphi^{-Az}(X, C_i) = O(\text{OPT})$. Hence, by Eq. (17) there also exists an estimator ζ_{C_i} , with positive constant probability, such that $\zeta_{C_i} = O(\text{OPT})$. In that case, we choose a set of centers C_A with $\varphi^{-10Az}(X, C_A) \leq 4\zeta_{C_A} = O(\text{OPT})$ with constant positive probability, by Eq. (18).

The estimator ζ_C for $\varphi^{-Az}(X, C)$ is constructed as follows. We sample $N = O((n/z)\log^2 n)$ points from X uniformly and independently at random with replacement. We denote the resulting multiset as X' . Then, we compute the value $\varphi^{-4Az(N/n)}(X', C)$ in time $O(|X'|k) = \tilde{O}(nk/z)$ and set $\zeta_C = \frac{n}{N}\varphi^{-4Az(N/n)}(X', C)$.

We define X_{in} as the closest Az points to the set C and $X_{out} = X \setminus X_{in}$. Let us split the points of X_{in} into $\log \Delta$ sets X_1, X_2, \dots such that $x \in X_i$ if $2^i \leq \varphi(x, C) \leq 2^{i+1}$. We say a set X_i is *big* if $|X_i| \geq z/\log \Delta$ and *small* otherwise. The intuition here is that either a set X_i is big enough so that the number of sampled points from X_i is concentrated around its expectation, or the set is small enough with respect to the number of outliers z we consider.

More precisely, for every big set X_i we can compute that $\mathbf{E}[|X_i \cap X'|] \geq \frac{z}{n \log \Delta} \cdot \Omega(\frac{n}{z} \log^2(n)) = \Omega(\log n)$, so by Theorem 7, we have

$$\begin{aligned} \mathbf{P}(\mathbf{E}[|X_i \cap X'|]/2 \leq |X_i \cap X'| \leq 2\mathbf{E}[|X_i \cap X'|]) & \\ \geq 1 - e^{-\Theta(|X_i \cap X'|)} & \\ = 1 - 1/\text{poly}(n). & \end{aligned} \quad (19)$$

Similarly, we can compute

$$\begin{aligned} \mathbf{P}(\mathbf{E}[|X_{out} \cap X'|]/2 \leq |X_{out} \cap X'| \leq 2\mathbf{E}[|X_{out} \cap X'|]) & \\ \leq 1 - e^{-\Theta(|X_{out} \cap X'|)} & \\ = 1 - 1/\text{poly}(n). & \end{aligned} \quad (20)$$

Furthermore, we have $|\cup_{i \text{ is small}} X_i| \leq \log(\Delta) \frac{z}{\log(\Delta)} = z$. As $\frac{z}{n} \cdot N = \Omega(\log^2 n)$, we can conclude that X' contains at most $2\frac{z}{n} \cdot N$ points from $\cup_{i \text{ is small}} X_i$ with probability $1 - 1/\text{poly}(n)$. We now condition on the events from Eq. (19) and Eq. (20) and we additionally condition on the event that X' contains at most $2\frac{z}{n} \cdot N$ points from $\cup_{i \text{ is small}} X_i$.

We first prove Eq. (17). We have $\zeta_C = \frac{n}{N}\varphi^{-4Az(N/n)}(X', C) \leq 4\varphi^{-Az}(X, C)$. By Eq. (20), we have $|X_{out} \cap X'| \leq 2Az(N/n)$, so we can label points in $X_{out} \cap X'$ as outliers. Furthermore, we can additionally label all points in $(\cup_{i \text{ is small}} X_i) \cap X'$ as outliers, as $|(\cup_{i \text{ is small}} X_i) \cap X'| \leq 2z(N/n)$. The number of sampled points from each big X_i is at most $2(N/n)|X_i|$ by Eq. (19) and the costs of points in X_i differ only by a 2-factor.

Next, we prove Eq. (18). Let z'_i be the number of outliers one needs to choose from $X_i \cap X'$ to optimize the cost $\varphi^{-4Az(N/n)}(X', C)$. Let us define a set of outliers X_{out}^A such that it contains the set X_{out} , all points from small sets X_i and $2z'_i(n/N)$ arbitrary points from each big set X_i . Then, $|X_{out}^A| \leq z + z + 2(4Az(N/n))(n/N) \leq 10Az$. To bound the cost $\varphi(X \setminus X_{out}^A, C)$, we bound the contribution of each set X_i . We have

$$\begin{aligned} & (|X_i| - 2z'_i(n/N)) \cdot 2^{i+1} \\ & \leq (2 \cdot \frac{n}{N}|X' \cap X_i| - 2z'_i \frac{n}{N}) 2^{i+1} & \text{Eq. (19)} \\ & \leq 4 \frac{n}{N} 2^i (|X' \cap X_i| - z'_i). \end{aligned}$$

Summing up these contributions, we get $\varphi^{-10Az}(X, C) \leq 4\frac{n}{N}\varphi^{-4Az(N/n)}$, as desired. \square

For the purpose of getting a set of centers that provides an $(O(1), O(1))$ -approximation in $\tilde{O}(nk^2/z)$ time, we next provide a simple adaptation of the ‘‘a clustering of a clustering is a clustering’’ result of (GMM⁺03).

Theorem 11. *Let X be an input set, OPT the optimal solution to the k -means with z outliers objective on X and Θ such that $OPT/(2z) \leq \Theta \leq OPT/z$. Suppose we have a set $Y, |Y| = O(k)$ such that $\tau_\Theta(X, Y) = O(OPT)$. Then, in time $\tilde{O}(nk^2/z)$ we can compute a set $C_{\mathcal{A}}, |C_{\mathcal{A}}| = k$, such that $\tau_\Theta(X, C_{\mathcal{A}}) = O(OPT)$ with positive constant probability.*

Proof. Let C^* be the clustering of an optimal solution. Recall that $\tau_\Theta(X, C^*) \leq OPT + z\Theta = O(OPT)$. We subsample $N = \Theta(\frac{nk}{z} \log n)$ points uniformly and independently with replacement from X , thus obtaining a multiset of points X' . For each point $x \in X'$ we compute its distance to Y . This is done in time $\tilde{O}(nk^2/z)$. For $y \in Y$ we define $B'(y)$ as the set of points $x \in X'$ for which y is the closest center and $B(y)$ as the set of points $x \in X$ for which y is the closest center. Moreover, we define $\hat{w}(y) = (n/N)|B'(y)|$ and $w(y) = |B(y)|$. Note that $\mathbf{E}[\hat{w}(y)] = w(y)$. We denote with y_x the point $y \in Y$ for which $x \in B(y)$. For each $y \in Y$ we either have $w(y) < z/k$, and then we call the respective set $B(y)$ *small*, or $w(y) \geq z/k$ and the respective set $B(y)$ is called *big*. For big sets we have $\mathbf{E}[|B'(y)|] \geq (N/n) \cdot z/k = \Omega(\log n)$. In this case, an application of Chernoff bounds (Theorem 7) gives that

$$w(y)/2 \leq \hat{w}(y) \leq 2w(y) \quad (21)$$

with probability $1 - e^{-\Omega(\mathbf{E}[|B'(y)|])} = 1 - 1/\text{poly}(n)$. We condition that this event happens for all $y \in Y$. We know that $\sum_{y \text{ small}} w(y) = O(z)$, so $\mathbf{E}[\sum_{y \text{ small}} \hat{w}(y)] = O(z)$ and by Markov’s inequality $\sum_{y \text{ small}} \hat{w}(y) = O(z)$ with constant probability. Again, we condition on this event.

We now run some $O(1)$ -approximate algorithm for k -means with penalties on the weighted instance $(Y, \hat{w}(y))$ that runs in $\tilde{O}(n'k)$ time for inputs of size n' (e.g., Algorithm 4 can be interpreted as such an algorithm by the easy part of the analysis captured by Lemma 9; the hard part of the analysis – Lemma 12 – then talks about the guarantees of the algorithm for k -means with outliers). As our instance has $n' = \tilde{O}(nk/z)$ points, the time complexity of the algorithm will be $\tilde{O}(nk^2/z)$. The algorithm returns a set $C_{\mathcal{A}}$ of k centers that we return. We have

$$\begin{aligned} & \tau_\Theta((Y, \hat{w}), C^*) & (22) \\ & \leq \sum_{y \text{ small}} \hat{w}(y)\Theta + \sum_{y \text{ big}} \hat{w}(y)\tau_\Theta(y, C^*) \\ & \leq O(z) \cdot \Theta + 2 \sum_{y \text{ big}} w(y)\tau_\Theta(y, C^*) & \text{Eq. (21)} \\ & \leq O(OPT) + 4 \sum_{y \text{ big}} \sum_{x \in B(y)} \tau_\Theta(y, x) + \tau_\Theta(x, C^*) & \text{Fact 2} \\ & \leq O(OPT) + O(\tau_\Theta(X, Y)) + O(\tau_\Theta(X, C^*)) = O(OPT). \end{aligned}$$

Hence,

$$\begin{aligned} & \tau_\Theta(X, C_{\mathcal{A}}) \\ & = \sum_{x \in X} \tau_\Theta(x, C_{\mathcal{A}}) \\ & \leq 2 \sum_{x \in X} (\tau_\Theta(x, y_x) + \tau_\Theta(y_x, C_{\mathcal{A}})) & \text{Fact 2} \\ & = 2\tau_\Theta(X, Y) + 2\tau_\Theta((Y, w), C_{\mathcal{A}}) \\ & \leq O(OPT) + 2 \sum_{y \text{ small}} w(y)\tau_\Theta(y, C_{\mathcal{A}}) + 2 \sum_{y \text{ big}} w(y)\tau_\Theta(y, C_{\mathcal{A}}) \\ & \leq O(OPT) + O(z) \cdot \Theta + 4 \sum_{y \text{ big}} \hat{w}(y)\tau_\Theta(y, C_{\mathcal{A}}) & \text{Eq. (21)} \\ & = O(OPT) + O(\tau_\Theta((Y, \hat{w}), C^*)) = O(OPT), & \text{Eq. (22)} \end{aligned}$$

as needed. \square

Remark 2. We believe, but do not prove, that refining the above proofs and using the results from Appendices C and F, we can refine Theorem 5 to achieve an $(O(1/\text{poly}(\varepsilon)), 1 + \varepsilon)$ -approximation algorithm with running time $\tilde{O}(nk^2/(z\text{poly}(\varepsilon)))$.

E. Lower bounds

In the following, we consider algorithms for the k -means/ k -median/ k -center with outliers problem in the general metric space query model. The algorithm is only required to output a set of k centers and does not need to explicitly declare points as outliers.

Theorem 12 (Formal version of Theorem 6). *Let \mathcal{A} denote a (randomized) algorithm for the k -means/ k -median/ k -center with outliers problem with a query complexity of $o(nk^2/z)$. Let $\alpha, \beta \in O(1)$ be arbitrary. Then, for large enough values k, z , and n such that $10000k \log k \leq z \leq \frac{1}{10000}n$, there exists a concrete input instance for the k -means/ k -median/ k -center problem with parameters k, z and n such that \mathcal{A} outputs an (α, β) -approximation with probability less than 0.5.*

If one does not assume that the ratio between the maximum and minimum distance is bounded, one can show that the statement even holds for an arbitrary multiplicative factor α . For simplicity, we assume in the proof that $\beta = 2$, but the proof works for an arbitrary constant β .

Proof. We roughly follow the proof of (Met02). Applying Yao's principle (Yao77), it suffices to provide a distribution over metric spaces such that any deterministic algorithm needs $\Omega(nk^2/z)$ queries to succeed with probability 0.5.

To that end, let k, z and n be given such that $10000k \log k \leq z \leq \frac{1}{10000}n$ and we assume that k is sufficiently large. Let $C = \{c_1, \dots, c_k\}$ be a set of points such that the distance between any two points in C is 1. Next, we define a probability distribution \mathcal{D} over C . Our random instance then consists of a multiset of n points $X = \{x_1, \dots, x_n\}$ with each point being sampled independently according to \mathcal{D} . More precisely, we call $S = \{c_1, \dots, c_{k/2}\}$ the set of *small* points and $B = \{c_{k/2+1}, \dots, c_k\}$ the set of *big* points (for simplicity, we assume that k is even). The probability of sampling each small point is set to $\frac{100z}{n \cdot k/2}$ and the probability of sampling each big point is set to $\frac{n-100z}{n \cdot k/2}$. This defines a valid probability distribution as $n \geq 100z$ and $\frac{k}{2} \cdot \frac{100z}{n \cdot k/2} + \frac{k}{2} \cdot \frac{n-100z}{n \cdot k/2} = 1$. Note that it can (and will) happen that we sample the same point twice, but the algorithm does not know it unless it asks the oracle \mathcal{M} , which for a query (x, y) returns the distance $d(x, y)$ of the two points in the constructed metric space. One could perturb this instance slightly in order to get an instance with a lower bound on the minimum distance between any two points. We refer to the multiset of points that are a clone of c_j as a small cluster if $j \in [k/2]$ and as a big cluster otherwise. We denote with $E_{i,j}$ the event that the i -th point is a clone of c_j .

We consider an arbitrary deterministic algorithm \mathcal{A} that asks at most $T = \frac{nk^2}{100000z}$ queries to the distance oracle \mathcal{M} . We refer to a query as *informative* if the distance between the two queried points is 0 and *uninformative* otherwise. We first start by proving a helper lemma.

Lemma 15. *Assume that among the first t queries of \mathcal{A} , at most $0.1k$ queries involved the i -th point and all of those queries were uninformative. Then, conditioned on the first t queries and the answer to those queries, the probability that the i -th point belongs to some arbitrary fixed small cluster with index $j_{\text{small}} \in [k/2]$ is at most $\frac{500z}{kn}$. Moreover, the statement still holds if we additionally condition on an arbitrary cluster assignment of all points, except the i -th one, that is consistent with the oracle answers to the first t queries.*

Proof. Let Q_t denote the event corresponding to the observed interaction between \mathcal{A} and the distance oracle restricted to the first t queries. To prove the lemma, we will not only condition on Q_t , but we additionally fix the randomness of all points except the i -th one in an arbitrary way. We denote the corresponding event by R_{-i} and we assume that $\mathbb{P}[Q_t \cap R_{-i}] > 0$. Among the first t queries, the i -th point was queried at most $0.1k$ times and all queries were uninformative. Hence, conditioned on Q_t and R_{-i} , there are at least $0.5k - 0.1k = 0.4k$ big clusters that the i -th point could still be contained in. Let j_{big} denote an arbitrary index of one of those big clusters. Using Bayes rule, we have

$$\begin{aligned}
 \mathbb{P}[E_{i,j_{small}} | Q_t \cap R_{-i}] &\leq \frac{\mathbb{P}[E_{i,j_{small}} | Q_t \cap R_{-i}]}{0.4k \cdot \mathbb{P}[E_{i,j_{big}} | Q_t \cap R_{-i}]} \\
 &= \frac{\mathbb{P}[Q_t \cap R_{-i} | E_{i,j_{small}}] \mathbb{P}[E_{i,j_{small}}]}{0.4k \cdot \mathbb{P}[Q_t \cap R_{-i} | E_{i,j_{big}}] \mathbb{P}[E_{i,j_{big}}]} \\
 &= \frac{\mathbb{P}[R_{-i} | E_{i,j_{small}}] \mathbb{P}[E_{i,j_{small}}]}{0.4k \cdot \mathbb{P}[R_{-i} | E_{i,j_{big}}] \mathbb{P}[E_{i,j_{big}}]} \\
 &= \frac{\mathbb{P}[E_{i,j_{small}}]}{0.4k \mathbb{P}[E_{i,j_{big}}]} \\
 &= \frac{100z}{0.4k \cdot (n - 100z)} \\
 &\leq \frac{500z}{kn}.
 \end{aligned}$$

□

Let $\mathcal{C}_{\mathcal{A}}$ denote the set of centers that \mathcal{A} outputs. To simplify the notation, we assume that $\mathcal{C}_{\mathcal{A}}$ is a set of k indices between 1 and n instead of k points. We define $\mathcal{C}_{\mathcal{A},small} = \mathcal{C}_{\mathcal{A}} \cap [k/2]$ as the set of indices corresponding to small clusters. The main ingredient of our lower bound argument is to show that $\mathbf{E}[|\mathcal{C}_{\mathcal{A},small}|]$ is small, namely at most $0.1k$. To that end, we partition $\mathcal{C}_{\mathcal{A},small} = \mathcal{C}_1 \sqcup \mathcal{C}_2 \sqcup \mathcal{C}_3$ into three different sets

$$\begin{aligned}
 \mathcal{C}_1 &:= \{i \in \mathcal{C}_{\mathcal{A},small} : \text{the } i\text{-th point was involved in less than} \\
 &\quad 0.1k \text{ queries, all being uninformative}\}, \\
 \mathcal{C}_2 &:= \{i \in \mathcal{C}_{\mathcal{A},small} : \text{the } i\text{-th point was involved in at least} \\
 &\quad 0.1k \text{ queries, the first } 0.1k \text{ of those being uninformative}\}, \\
 \mathcal{C}_3 &:= \{i \in \mathcal{C}_{\mathcal{A},small} : \text{Among the first } 0.1k \text{ queries} \\
 &\quad \text{the } i\text{-th point was involved with, at least one was informative}\}.
 \end{aligned}$$

We prove that the expected size of all three sets is small. We start by bounding $\mathbf{E}[|\mathcal{C}_1|]$. To that end, let Q denote the event corresponding to some (fixed) complete observed interaction between \mathcal{A} and the oracle. As \mathcal{A} is deterministic, conditioned on Q , the set $\mathcal{C}_{\mathcal{A}}$ is completely determined. Consider some arbitrary $i \in \mathcal{C}_{\mathcal{A}}$ such that the i -th point was involved in at most $0.1k$ queries, all being uninformative. [Lemma 15](#) together with a union bound over all small clusters implies that

$$\mathbb{P}[\text{the } i\text{-th point belongs to a small cluster} \mid Q] \tag{23}$$

$$\leq 0.5k \cdot \frac{500z}{kn} = \frac{250z}{n}. \tag{24}$$

Hence, using linearity of expectation, we obtain $\mathbf{E}[|\mathcal{C}_1|] \leq k \cdot \frac{250z}{n} \leq 0.025k$.

Next, we bound the expected size of \mathcal{C}_2 . Note that there can be at most $2 \cdot \frac{nk^2/(100000z)}{0.1k} = \frac{nk}{5000z}$ points that are involved in at least $0.1k$ queries, the first $0.1k$ of those being uninformative. Among those points, the expected fraction of points belonging to small clusters is at most $\frac{250z}{n}$. This is again a consequence of [Eq. \(23\)](#), as at the moment the i -th point was queried exactly $0.1k$ times, all queries being uninformative, the probability that the point belongs to a small cluster is at most $\frac{250z}{n}$. Hence, we can conclude that $\mathbf{E}[|\mathcal{C}_2|] \leq \frac{250z}{n} \cdot \frac{nk}{5000z} = 0.05k$.

Thus, it remains to bound $\mathbf{E}[|\mathcal{C}_3|]$. To that end, for $t \in [T]$, let X_t denote the indicator variable for the following event: The t -th query of \mathcal{A} is informative, involves a point from a small cluster that was queried less than $0.1k$ times before and all those previous queries were uninformative. Again, [Lemma 15](#) implies that $\mathbf{E}[X_t] \leq \frac{500z}{k \cdot n}$. As $|\mathcal{C}_3| \leq 2 \cdot \sum_{t \in [T]} X_t$, linearity of expectation implies that

$$\mathbf{E}[|\mathcal{C}_3|] \leq 2 \frac{nk^2}{100000z} \cdot \frac{500z}{k \cdot n} \leq 0.01k.$$

Putting things together, we obtain that $\mathbf{E}[|\mathcal{C}_{\mathcal{A},small}|] \leq 0.1k$. By Markov's inequality, this implies that $\mathbf{P}(|\mathcal{C}_{\mathcal{A},small}| \geq 0.3k) \leq 1/3$. As the expected size of each small cluster is $\frac{200z}{k} \geq 200 \log k$, a Chernoff Bound followed by a Union Bound implies that each small cluster contains at least $100z/k$ points with a probability of at least 0.9. Hence, with a probability of at least $0.9 - 1/3 > 0.5$, each small cluster contains at least $100z/k$ points and \mathcal{A} outputs at most $0.3k$ centers belonging to small clusters. As $(0.5k - 0.3k) \cdot \frac{100z}{k} > 2z$, this implies that the clustering cost with respect to $\mathcal{C}_{\mathcal{A}}$ is not zero, unlike the clustering cost of an optimal solution. This finishes the proof. \square

Remark 3. Note that the same construction can be used to give an $\Omega(k^2/\varepsilon)$ lower bound for k -means algorithms that output a solution C with $\varphi(X, C) = O(\varphi(X, C^*) + \varepsilon\varphi(X, \mu(X)))$. This shows that the complexity of the AFK-MC² algorithm (BLHK16a) is essentially the best possible. We omit details.

F. Distributed algorithms

In this section, we show how to adapt the classical streaming algorithm of (GMM⁺03) and the popular distributed algorithm k -means|| to the setting with outliers by extending Theorem 8. In Section F.1, we show two ways of, roughly speaking, distributing Theorem 8. Then, in Section F.2, we show how this set can be used to get distributed algorithms that output $(1 + \varepsilon)z$ outliers and achieve an $O(1/\varepsilon)$ or even $O(1)$ -approximation.

The most natural distributed model for the following algorithms is the coordinator model. In this model, the data $X = X_1 \sqcup X_2 \sqcup \dots \sqcup X_m$ is evenly split across m machines. In a 1-round distributed algorithm such as Algorithm 5, the machines perform some computation on their data, and send the result to the coordinator who computes the final clustering. In a t -round distributed algorithm such as Algorithm 6, there are t rounds of communication between machines and the coordinator.

F.1. Distributed algorithms with bicriteria guarantee

In this section, we start by presenting Algorithm 5, a simple variant of the algorithm of (GMM⁺03). In this algorithm, each machine runs Algorithm 2 with oversampling and sends the resulting clustering, together with some additional information, to the coordinator. Analysis of this algorithm follows from our Theorem 8.

Algorithm 5 Overseeding from Guha et al. (GMM⁺03)

Require data X , threshold $\text{OPT}/(2\varepsilon) \leq \Theta \leq \text{OPT}/\varepsilon$

- 1: split X arbitrarily into sets X_1, X_2, \dots, X_m .
 - 2: **for** $j \leftarrow 1, 2, \dots, m$ **in parallel do**
 - 3: $Y_j \leftarrow$ Algorithm 2 with $X_{A2} = X_j$, $\ell_{A2} = \tilde{O}(k/\varepsilon)$, and $\Theta_{A2} = \Theta$.
 - 4: For each $y \in Y_j$, let $w(y)$ be the number of points $x \in X_j$, $\tau_{\Theta}(x, Y_j) < \Theta$, such that $y = \arg \min_{y' \in Y_j} \tau_{\Theta}(x, y')$ with ties handled arbitrarily. Let $X_{out,j}$ be the set of points $x \in X_j$ with $\tau(x, Y_j) = \Theta$.
 - 5: **end for**
 - 6: Each site j sends (Y_j, w) and $|X_{out,j}|$ to the coordinator, who computes the weighted set $(Y, w) = \bigcup_{j=1}^m (Y_j, w)$ and the cardinality of $X_{out} = \bigcup_j X_{out,j}$ as $|X_{out}| = \sum_j |X_{out,j}|$.
-

Here, by weighted set (Y, w) we mean a set $Y = \{y_1, y_2, \dots, y_{|Y|}\}$ together with a weight function $w : \{1, 2, \dots, |Y|\} \rightarrow \mathbb{N}$. Set operations can be naturally extended to the weighted setting and in the next subsection we observe that k -means algorithms too.

Theorem 13. Let $\text{OPT}/(2\varepsilon z) \leq \Theta \leq \text{OPT}/(\varepsilon z)$. Suppose we run Algorithm 5 on the m machines with $\ell = O(k/\varepsilon \log(m/\delta))$. Then with probability $1 - \delta$ we have

$$\sum_{j=1}^m \tau(X_j \cap X_{in}^*, Y_j) = O(\text{OPT}).$$

Proof. Let $z_j = |X_j \cap X_{out}^*|$ and recall C^* is the optimal solution for the whole instance X . By Theorem 8 we have with

probability at least $1 - m \cdot \delta/m = 1 - \delta$ that for every j

$$\begin{aligned} \tau(X_j \cap X_{in}^*, Y_j) &= O\left(\inf_{C, |C|=k} \tau(X_j \cap X_{in}^*, C) + \varepsilon z_j \Theta\right) \\ &= O(\tau(X_j \cap X_{in}^*, C^*) + \varepsilon z_j \Theta). \end{aligned}$$

This implies that we have

$$\begin{aligned} \sum_{j=1}^m \tau(X_j \cap X_{in}^*, Y_j) &= \sum_{j=1}^m O(\tau(X_j \cap X_{in}^*, C^*) + \varepsilon z_j \Theta) \\ &= O(\tau(X_{in}^*, C^*) + \varepsilon z \Theta) = O(\text{OPT}). \end{aligned}$$

□

We leave the description of the algorithm run by the coordinator to [Section F.2](#) and now we show a different way of getting the weighted set (Y, w) with essentially the same guarantees, by adapting the k -means|| algorithm.

The k -means|| algorithm We now show a simple adaptation of the k -means|| algorithm ([BMV⁺12](#)): a popular distributed variant of k -means++. In k -means||, the goal is to get $\tilde{O}(k)$ centers that provide a constant approximation guarantee in a small number of distributed steps. This is done by changing [Algorithm 2](#) as follows: in each round we sample k points instead of just one point proportional to the cost, but the algorithm is run only for $O(\log n)$ steps. Our adaptation of k -means|| algorithm to outliers is below.

Algorithm 6 k -means|| overseeding

Require data X , # rounds t , sampling factor ℓ , threshold Θ

- 1: Uniformly sample $x \in X$ and set $Y = \{x\}$.
 - 2: **for** $i \leftarrow 1, 2, \dots, t$ **do**
 - 3: $Y' \leftarrow \emptyset$
 - 4: **for** $x \in X$ **do**
 - 5: Add x to Y' with probability $\min\left(1, \frac{\ell \tau_{\Theta}(x, Y)}{\tau_{\Theta}(X, Y)}\right)$
 - 6: **end for**
 - 7: $Y \leftarrow Y \cup Y'$
 - 8: **end for**
 - 9: For each $y \in Y$, let $w(y)$ be the number of points $x \in X_j$ with $\tau_{\Theta}(x, Y) < \Theta$ and such that $y = \arg \min_{y' \in Y} \tau_{\Theta}(x, y')$ with ties handled arbitrarily. Let X_{out} be the set of points x with $\tau(x, Y) = \Theta$.
 - 10: **Return** $(Y, w), |X_{out}|$
-

We now prove an analogue of [Theorem 13](#). Again, it can be seen as an extension of our basic result [Theorem 8](#).

Theorem 14. *Let $\Theta \in [OPT/(2\varepsilon), OPT/\varepsilon]$ be arbitrary. Suppose we run [Algorithm 6](#) for $t = O(\log(n/\varepsilon))$ rounds with $\ell = O((k/\varepsilon) \log n)$ to obtain output Y . Then, with probability at least $1 - 1/\text{poly}(n)$, we have that*

$$\tau_{\Theta}(X_{in}^*, Y) = O(\text{OPT}).$$

The following proof is a simple adaptation of the analysis of k -means|| by [Rozhon \(Roz20\)](#).

Proof. Let $Y_0 \subseteq Y_1 \subseteq \dots \subseteq Y_t$ be the cluster centers that are gradually built by [Algorithm 6](#). We have $Y_0 = \{y\}$ for a point y picked uniformly at random. Whatever point we picked, we have $\tau(X, Y_0) \leq n\Theta = \text{poly}(n/\varepsilon)$, as we assume $\Delta = \text{poly}(n)$. We call a cluster X_i^* *unsettled* in iteration j if $\tau(X_i^*, Y_j) > 10\tau(X_i^*, C^*)$. We define $\tau_U(X, Y_j)$ as

$$\tau_U(Y, Y_j) = \sum_{i, X_i^* \text{ unsettled in iter. } j} \tau(X_i^*, Y_j).$$

We will now prove that with probability $1 - 1/\text{poly}(n)$ we have

$$\tau_U(X, Y_{j+1}) \leq \frac{1}{2}\tau_U(X, Y_j) + \text{OPT}. \quad (25)$$

To prove Eq. (25), fix an iteration j . If $\tau_U(X, Y_j) \leq \text{OPT}$, the claim certainly holds, so assume the opposite. We split unsettled clusters into two groups: a cluster X_i^* is called *heavy* if $\tau(X_i^*, Y_j) \geq \tau_U(X, Y_j)/(2k)$ and *light* otherwise. The probability that we make a heavy cluster X_i^* settled can be bounded as follows. Sampling a random point from X_i^* proportional to $\tau(\cdot, Y_j)$ makes X_i^* settled with probability at least $1/5$ by Corollary 1. Hence, there is a subset $Z_i \subseteq X_i^*$ with $\tau(Z_i, Y_j) \geq \tau(X_i^*, Y_j)/5$ such that sampling a point from Z_i makes X_i^* settled. If Z_i contains a point x with $\tau(x, Y_j) \geq \frac{\tau(X_i^*, Y_j)}{\ell}$, we sample x with probability 1 and this also makes X_i^* settled. Otherwise,

$$\begin{aligned} & \text{P}(X_i^* \text{ does not get settled}) \\ & \leq \prod_{x \in Z_i} (1 - \ell \cdot \tau(x, Y_j) / \tau(X, Y_j)) \\ & \leq \exp(-O(k \log n / \varepsilon)) \sum_{x \in Z_i} \tau(x, Y_j) / \tau(X, Y_j) && 1 + x \leq e^x \\ & \leq \exp(-O(k(\log n) / \varepsilon)) \cdot \frac{1}{5} \tau(X_i^*, Y_j) / \tau(X, Y_j) \\ & \leq \exp(-O((\log n) / \varepsilon)) \cdot \frac{1}{10} \tau_U(X, Y_j) / \tau(X, Y_j) && X_i^* \text{ is heavy} \\ & \leq \exp\left(-O((\log n) / \varepsilon) \cdot \frac{1}{10} \frac{\tau_U(X, Y_j)}{\tau_U(X, Y_j) + 10\text{OPT} + z\Theta}\right) \\ & \text{each point is unsettled, settled, or an outlier} \\ & \leq \exp\left(-O((\log n) / \varepsilon) \cdot \frac{1}{10} \frac{\text{OPT}}{11\text{OPT} + \text{OPT}/\varepsilon}\right) \\ & \qquad \qquad \qquad \tau_U(X, Y_j) \geq \text{OPT} \\ & = 1/\text{poly}(n). \end{aligned}$$

Hence, every heavy cluster X_i^* gets settled with probability $1 - 1/\text{poly}(n)$. We now condition on that event.

Note that $\sum_{X_i^* \text{ light}} \tau(X_i^*, Y_j) \leq k \cdot \tau_U(X, Y_j)/(2k) = \tau_U(X, Y_j)/2$. Hence, we get

$$\tau_U(X, Y_j) - \tau_U(X, Y_{j+1}) \geq \sum_{X_i^* \text{ heavy}} \tau(X_i^*, Y_j) \geq \tau_U(X, Y_j)/2$$

as needed to finish the proof of Eq. (25). Now we can apply Eq. (25) $t = O(\log(n/\varepsilon))$ times together with $\tau(X, Y_0) \leq \text{poly}(n/\varepsilon)$ to conclude that

$$\begin{aligned} & \tau(X_{in}^*, Y_t) \\ & \leq O(\text{OPT}) + (1/2)^t \text{poly}(n/\varepsilon) + \text{OPT} \cdot \sum_{j=0}^t (1/2)^j \\ & = O(\text{OPT}), \end{aligned}$$

with probability $1 - 1/\text{poly}(n)$. □

F.2. Constructing the Final Clustering

In this subsection, we show how the output of Algorithms 5 and 6, that is, a weighted set of centers (Y, w) , together with the number of found outliers $|X_{out}|$, can be used to get $(O(1), 1 + \varepsilon)$ -approximation algorithms, via Theorems 13 and 14. First, in Theorem 15, we only show how to get an $(O(1/\varepsilon), 1 + \varepsilon)$ -approximation guarantee by simply running Algorithm 4 on

the weighted data with the parameter of number of outliers set to roughly $z - |X_{out}|$. The advantage of [Theorem 15](#) is its simplicity (we implement a variant of it in [Section 7](#)) and speed.

Next, in [Theorem 16](#) we refine the reduction to get an $(O(1), 1 + \varepsilon)$ -approximation guarantee. This result is interesting from a theoretical perspective, as we explained in [Section 5](#). As a subroutine, the coordinator needs to run some $(O(1), 1 + \varepsilon)$ -approximation algorithm, i.e., linear programming based algorithms ([Che08](#); [KLS18](#)), so the resulting distributed algorithm seems not practical. Also, we need to refine [Algorithms 5](#) and [6](#) so that they pass somewhat more information than just (Y, w) and $|X_{out}|$.

Theorem 15 (Adaptation of ([GMM⁺03](#))). *Let $\varepsilon \in [0, 1]$ be arbitrary and Θ be such that $OPT/(2\varepsilon z) \leq \Theta \leq OPT/(\varepsilon z)$. Let (Y, w) denote the weighted point set that [Algorithm 6](#) and [Algorithm 5](#) compute, respectively. Furthermore, assume that $\tau_\Theta(X_{in}^*, Y) \leq \alpha OPT$ in case of [Algorithm 6](#), and $\sum_j \tau_\Theta(X_j \cap X_{in}^*, Y_j) \leq \alpha OPT$, $Y = \bigcup Y_j$, in case of [Algorithm 5](#), respectively. Now, suppose we then run a $(\beta, 1 + \varepsilon)$ -approximation algorithm \mathcal{A} for the weighted k -means with outliers formulation on the weighted instance (Y, w) with $z^A := z - |X_{out}| + 2\alpha\varepsilon z$ outliers and let C denote the resulting set of k centers. Then,*

$$\varphi^{-(1+5\alpha\varepsilon)z}(X, C) = O((\alpha\beta + 1/\varepsilon)OPT).$$

Here, we define weighted k -means with outliers as the following problem: for the weighted input (Y, w) , where each $w(y)$ is an integer, we want to choose a set of k centers C and a weight vector $w_{in}(y)$ for any y such that $w_{in}(y)$ is a non-negative integer, for $w_{out}(y) = w(y) - w_{in}(y)$ we have $\sum_{y \in Y} w_{out}(y) \leq z^A$ and the sum $\sum_{y \in Y} w_{in}(y)\varphi(y, C)$ is minimized.

Note that algorithms for k -means with outliers can be seen as algorithms for weighted k -means with outliers by viewing weighted points $(y, w(y))$ as $w(y)$ points on the same location (our assumption that pairwise distances of input points are at least one, is in our case only for simplicity of exposition). Moreover, sampling based algorithms such as [Algorithm 3](#) can be implemented in time proportional not to $\sum_y w(y)$, but $|Y| = \tilde{O}(k)$ by additionally weighting the sampling distribution by the weight of input points.

Plugging in our [Algorithm 3](#) as \mathcal{A} in [Theorem 15](#) hence yields a distributed algorithm with approximation factor $O(\alpha\beta + 1/\varepsilon) = O(1/\varepsilon + 1/\varepsilon) = O(1/\varepsilon)$ that outputs $(1 + O(\varepsilon))z$ outliers. Moreover, the computational complexity of the coordinator is $\tilde{O}(k^2/\varepsilon^2)$ if we use [Algorithm 6](#) and $\tilde{O}(mk^2/\varepsilon^2)$ if we use [Algorithm 5](#).

Proof. We start by showing that $|X_{in}^* \cap X_{out}| \leq 2\alpha\varepsilon z$. That is, the number of points that [Algorithm 5](#) and [Algorithm 6](#), respectively, declare as outliers in the first phase of the algorithm, but that are actually true inliers with respect to a given optimal solution is small. For [Algorithm 6](#), we have

$$|X_{in}^* \cap X_{out}| \leq \frac{\tau_\Theta(X_{in}^*, Y)}{\Theta} \leq \frac{\alpha OPT}{OPT/(2\varepsilon z)} = 2\alpha\varepsilon z,$$

and for [Algorithm 5](#), we have

$$|X_{in}^* \cap X_{out}| \leq \sum_j \frac{\tau_\Theta(X_j \cap X_{in}^*, Y_j)}{\Theta} \leq \frac{\alpha OPT}{OPT/(2\varepsilon z)} = 2\alpha\varepsilon z,$$

as desired. In particular, $|X_{out}| = |X_{out} \cap X_{out}^*| + |X_{out} \cap X_{in}^*| \leq z + 2\alpha\varepsilon z$. Hence, $z^A = z - |X_{out}| + 2\alpha\varepsilon z \geq 0$ and therefore algorithm \mathcal{A} is run on a legal instance. Let $X_{in} := X \setminus X_{out}$. Notice that for [Algorithm 6](#), we have

$$\varphi(X_{in}^* \cap X_{in}, Y) = \tau_\Theta(X_{in}^* \cap X_{in}, Y) \leq \alpha OPT, \quad (26)$$

and for [Algorithm 5](#), we have

$$\sum_j \varphi(X_{in}^* \cap X_{in} \cap X_j, Y_j) \quad (27)$$

$$= \sum_j \tau_\Theta(X_{in}^* \cap X_{in} \cap X_j, Y_j) \leq \alpha OPT. \quad (28)$$

Let us call each set of points $B(y) \subseteq X$ that [Algorithm 5](#) or [Algorithm 6](#) groups around $y \in Y$ a *blob* and for a given $x \in X_{in}$, we denote with y_x the point with $x \in B(y_x)$. We define (as the respective algorithm also does) $w(y) = |B(y)|$. Moreover, we define $w_{in}^*(y) = |B(y) \cap X_{in}^*|$ and $w_{out}^*(y) = |B(y) \cap X_{out}^*|$.

Note that splitting the weights w into w_{in}^* and w_{out}^* gives us a natural upper bound on the cost of the optimal solution on the instance Y with z^A outliers: On one hand, we have

$$\begin{aligned}
 \sum_{y \in Y} w_{out}^*(y) &= |X_{out}^* \cap X_{in}| & (29) \\
 &= |X_{out}^*| - |X_{out}^* \cap X_{out}| \\
 &= z - |X_{out}| + |X_{out} \cap X_{in}^*| \\
 &\leq z - |X_{out}| + 2\alpha\epsilon z \\
 &= z^A.
 \end{aligned}$$

Thus, if we label the weighted set (Y, w_{out}^*) as outliers, at most z^A points are labeled as outliers. On the other hand, we will now bound $\varphi((Y, w_{in}^*), C^*)$, where C^* is the optimum clustering for the original problem defined on X .

For any $y \in Y$, $x \in X_{in}^* \cap B(y)$, and $c_x^* = \arg \min_{c^* \in C^*} \varphi(x, c^*)$ we have

$$\varphi(y, C^*) \leq \varphi(y, c_x^*) \leq 2\varphi(y, x) + 2\varphi(x, c_x^*), \quad (30)$$

where the second inequality is due to [Fact 2](#). Hence,

$$\varphi((Y, w_{in}^*), C^*) \quad (31)$$

$$= \sum_{y \in Y} w_{in}^*(y) \cdot \varphi(y, C^*) \quad (32)$$

$$\begin{aligned}
 &= \sum_{y \in Y} \sum_{x \in X_{in}^* \cap B(y)} \varphi(y, C^*) \\
 &\leq \sum_{y \in Y} \sum_{x \in X_{in}^* \cap B(y)} 2\varphi(y, x) + 2\varphi(x, c_x^*)
 \end{aligned}$$

Eq. (30)

$$\leq 2 \left(\sum_{x \in X_{in}^* \cap X_{in}} \varphi(y_x, x) \right) + 2 \left(\sum_{x \in X_{in}^*} \varphi(x, c_x^*) \right)$$

$$\leq 2\alpha\text{OPT} + 2\text{OPT}$$

Eqs. (26) and (27)

$$= O(\alpha\text{OPT}).$$

Now we consider the output C^A and w_{out}^A of \mathcal{A} , i.e., for each $y \in Y$ the algorithm \mathcal{A} decides which integer weight w_{out}^A of y is labeled as outlier. Define $w_{in}^A(y) = w(y) - w_{out}^A(y)$. This solution of instance (Y, w) with $\sum_{y \in Y} w_{out}^A(y)$ outliers naturally defines a solution of the original instance X as follows. We leave the set of centers C^A the same and whenever $w_{out}^A(y)$ is nonzero, we label $w_{out}^A(y)$ arbitrary points in $B(y)$ as outliers and call this set X_{out}^A (note that the algorithm itself does not need to explicitly label the outliers, it suffices to prove that there is a labelling). Then we define points from $X_{out} \cup X_{out}^A$ as outliers. The number of points we label as outliers is bounded by

$$\begin{aligned}
 |X_{out}| + \sum_{y \in Y} w_{out}^A(y) &\leq |X_{out}| + (1 + \epsilon)z_A & (33) \\
 &= |X_{out}| + (1 + \epsilon)(z - |X_{out}| + 2\alpha\epsilon z) \\
 &\leq (1 + 5\alpha\epsilon)z
 \end{aligned}$$

Thus, it remains to bound the cost of the set $X \setminus (X_{out} \cup X_{out}^A)$ with respect to C^A . We have

$$\begin{aligned} & \varphi(X \setminus (X_{out} \cup X_{out}^A), C^A) \\ &= \sum_{y \in Y} \sum_{x \in B(y) \setminus X_{out}^A} \varphi(x, C^A) \\ &\leq 2 \sum_{y \in Y} \sum_{x \in B(y)} \varphi(x, y_x) + 2 \sum_{y \in Y} \sum_{x \in B(y) \setminus X_{out}^A} \varphi(y_x, C^A) \end{aligned}$$

Fact 2

$$\begin{aligned} &= 2 \sum_{y \in Y} \sum_{x \in B(y) \cap X_{in}^*} \varphi(x, y_x) + 2 \sum_{y \in Y} \sum_{x \in B(y) \cap X_{out}^*} \varphi(x, y_x) \\ &+ 2 \sum_{y \in Y} \sum_{x \in B(y) \setminus X_{out}^A} \varphi(y_x, C^A) \\ &\leq 2\alpha \text{OPT} + 2\varphi(X_{in} \cap X_{out}^*, Y) + 2 \sum_{y \in Y} w_{in}^A(y) \varphi(y, C^A) \end{aligned}$$

Eq. (26), Eq. (27)

$$\leq 2\alpha \text{OPT} + 2z\Theta + 2 \sum_{y \in Y} w_{in}^A(y) \varphi(y, C^A)$$

$$|X_{out}^*| = z, x \in X_{in} \Rightarrow \varphi(x, Y) \leq \Theta$$

$$\leq 2\alpha \text{OPT} + 2(1/\varepsilon)\text{OPT} + 2\beta\varphi(Y, w_{in}^*, C^*)$$

\mathcal{A} is β -approximation

$$\leq O((1/\varepsilon + \alpha\beta)\text{OPT}).$$

Eq. (31)

□

Better construction Here we sketch a somewhat more elaborate construction that enables us to lose only a constant factor in approximation, instead of $O(1/\varepsilon)$. Below, we assume the existence of a $(\beta, 1 + \varepsilon)$ -approximation algorithm that works in what we call an *almost-metric space*, which is defined as a classical metric space, but without the axiom $d(x, y) = 0 \Leftrightarrow x = y$. That is, even when an algorithm chooses some point x as a cluster center, the clustering cost of x might still be non-zero. Our algorithms work in this more general setting and we believe that the algorithm of (BERS19) too. In any case, note that the problem of k -means with outliers in an almost metric space can be reduced to the same problem in a metric space by splitting each vertex x into $2k$ clones x_1, \dots, x_{2k} and for $i \neq j$ defining $d'(x_i, x_j) = d(x, x)$. This increases the number of points by a $2k$ -factor and we lose only a factor of 2 in our approximation guarantee. As in the previous construction, we moreover assume a weighted version of the problem, where for a given integer weight $w(x)$ one is allowed to output an integer $0 \leq w_{out}(x) \leq w(x)$ and then only $w(x) - w_{out}(x)$ weight is used in the computation of the cost.

Theorem 16. *Let $\varepsilon \in (0, 1]$ be arbitrary. Suppose that there is a $(\beta, 1 + \varepsilon)$ -approximation algorithm \mathcal{A} that solves the weighted variant of k -means with outliers in an almost-metric space in polynomial time. Then, there is a distributed polynomial-time $(O(\beta), 1 + O(\varepsilon))$ -approximation algorithm in the coordinator model with communication $\tilde{O}(k/\varepsilon)$ per site that succeeds with positive constant probability.*

Proof. Assume that Θ satisfies $\frac{\text{OPT}}{2\varepsilon z} \leq \Theta \leq \frac{\text{OPT}}{\varepsilon z}$. We later discuss how to "guess" OPT in this distributed setting. The algorithm needed is a variant of Algorithm 5, so we only describe the necessary changes. When the variant of Algorithm 5 aggregates all points to the closest center in Y_j , it will not just compute for each y its weight $w(y)$, i.e., for how many points in $X_j \setminus X_{out}$ the point y is the closest, but it iterates over all these points $x_1, x_2, \dots, x_{w(y)}$ and rounds the distances $d(x_i, y)$ down to the closest power of two. Then, the additional information sent to the leader about y is not just $w(y)$, but also the list $w_0(y), w_1(y), w_2(y), \dots$, where $w_k(y)$ is the number of points $x \in B(y)$ with distance $d(x, y)$ rounded down to 2^k . For any $y \in Y_j$ and k , we define $B_k(y)$ as the set of points in $X_j \setminus X_{out}$ for which y is the closest point in Y_j and $d(x, y)$ is rounded down to 2^k . As before, for a given $x \in X_{in} := X \setminus X_{out}$, y_x denotes the point y for which $x \in B(y)$.

The instance (Y', w') the leader is going to construct will not be just (Y, w) , but the leader creates a new almost-metric space \mathcal{M}' that includes the original input metric space \mathcal{M} that is only defined for points of Y together with the original metric, but moreover, for each y and $0 \leq k \leq \log \Delta$, it contains a point y^k with $d_{\mathcal{M}'}(y, y^k) = 2^k$. Now, imagine a weighted graph with the set of vertices being equal to the points in the almost-metric space and there is an edge between two points if we explicitly stated the distance between the two points and the weight of the edge is equal to that distance. Then, the distance between each pair of vertices is just equal to the length of the shortest path between the two vertices in the weighted graph. In fact, we also define $d_{\mathcal{M}'}(y^k, y^k) = 2 \cdot 2^k$ (which is not possible in a metric space) and only then the leader runs the $(\beta, 1 + \varepsilon)$ -approximation algorithm \mathcal{A} in \mathcal{M}' . We note that the defined distances satisfy the triangle inequality.

As in [Theorem 15](#), we assume that $\sum_j \tau_{\Theta}(X_j \cap X_{in}^*, Y_j) \leq \alpha \text{OPT}$. We run \mathcal{A} on \mathcal{M}' (we will use φ' and d' when talking about (squared) distances in \mathcal{M}') with the number of outliers being set to $z^{\mathcal{A}} = z - |X_{out}| + 2\alpha\varepsilon z$ and get as output a set of centers $\mathcal{C}^{\mathcal{A}}$ and a set of outliers $(Y, w_{out}^{\mathcal{A}})$ with $\sum_{y \in Y} w_{out}^{\mathcal{A}}(y) \leq (1 + \varepsilon)z^{\mathcal{A}}$.

As in the proof of [Theorem 15](#), we would now like to argue that the optimal solution in the almost-metric space has a cost of at most $O(\alpha \text{OPT})$ by considering the optimal set of centers C^* for the original problem. However, there is one small technical issue. Namely, the points in C^* might not be contained in the almost-metric space. To remedy this situation, we do the following: We naturally extend the almost-metric space to also include all the points in C^* and prove that the defined distances still satisfy the triangle inequality. Next, we show that in this extended metric space, the cost of the optimal solution (with outliers defined appropriately) has a cost of $O(\alpha \text{OPT})$. By using the uniform sampling lemma ([Lemma 3](#)), this implies that there also exists a solution in the original almost-metric space (without the points in C^*) with a cost of at most $4 \cdot O(\alpha \text{OPT}) = O(\alpha \text{OPT})$, as desired.

We start by extending the almost-metric space to points in C^* . For each $c^* \in C^*$ and $y \in Y$, we define $d_{\mathcal{M}'}(y, c^*) = d(y, c^*)$. That is, the distances between points in Y and C^* are simply equal to their original distances. The distance between $c^* \in C^*$ and y^k is defined as $d_{\mathcal{M}'}(c^*, y^k) = d_{\mathcal{M}'}(c^*, y) + d_{\mathcal{M}'}(y, y^k) = d_{\mathcal{M}'}(c^*, y) + 2^k$. This extension still satisfies the triangle inequality, which more or less directly follows from the fact that the original metric satisfies the triangle inequality. Next, we define a solution for the weighted k -means with $z^{\mathcal{A}}$ outliers objective in this extended almost metric-space, whose cost gives us an upper bound of $O(\alpha \text{OPT})$ for the optimal solution. We consider the optimal set of centers C^* together with the set of outliers X_{out}^* . We set $w_{out}^*(y^k) = |B_k(y) \cap X_{out}^*|$, $w_{out}^*(y) = |\{y\} \cap X_{out}^*|$, $w_{in}^*(y^k) = w_k(y) - w_{out}^*(y^k)$ and $w_{in}^*(y) = w_k(y) - w_{out}^*(y)$. Splitting the weights w into w_{in}^* and w_{out}^* together with the set of centers C^* gives us a natural upper bound (up to a factor of 4, as mentioned before) on the cost of the optimal solution on the instance (Y', w') with $z^{\mathcal{A}}$ outliers. First, one needs to verify that the number of declared outliers $\sum_{y \in Y} (w_{out}^*(y) + \sum_k w_{out}^*(y^k))$ in the solution is at most $z_{\mathcal{A}}$. This follows in the exact same way as in the proof of [Theorem 15](#) (cf. [Eq. \(29\)](#)). Thus, it remains to bound $\varphi'((Y', w_{in}^*), C^*)$.

Let $x \in B_k(y)$ be arbitrary for some k and $y \in Y$. Moreover, let $c_x^* := \arg \min_{c^* \in C^*} \varphi(x, c^*)$. We have

$$\begin{aligned}
 & \varphi'(y^k, C^*) & (34) \\
 & \leq 2\varphi'(y^k, y) + 2\varphi'(y, C^*) \\
 & = 2\varphi'(y^k, y) + 2\varphi(y, C^*) \\
 & \leq 2\varphi'(y^k, y) + 4\varphi(y, x) + 4\varphi(x, c_x^*) \\
 & \leq 6\varphi(y, x) + 4\varphi(x, c_x^*) & \varphi'(y^k, y) \leq \varphi(y, x),
 \end{aligned}$$

where we repeatedly used [Fact 2](#). Hence,

$$\varphi'((Y', w_{in}^*, C^*)) \tag{35}$$

$$= \sum_{y \in Y} w_{in}^*(y) \cdot \varphi'(y, C^*) + \sum_{y^k \in Y'} w_{in}^*(y^k) \cdot \varphi'(y^k, C^*) \tag{36}$$

$$= \sum_{y \in Y \cap X_{in}^*} \varphi'(y, C^*) + \sum_{y^k \in Y'} \sum_{x \in X_{in}^* \cap B_k(y)} \varphi'(y^k, C^*) \tag{37}$$

$$\leq \sum_{y \in Y \cap X_{in}^*} \varphi'(y, C^*) + \sum_{y^k \in Y'} \sum_{x \in X_{in}^* \cap B_k(y)} 6\varphi(y, x) + 4\varphi(x, c_x^*)$$

Eq. (34)

$$\leq 6 \left(\sum_{x \in X_{in}^* \cap X_{in}} \varphi(y_x, x) \right) + 4 \left(\sum_{x \in X_{in}^*} \varphi(x, c_x^*) \right)$$

$$\leq 6O(\alpha \text{OPT}) + 4\text{OPT}$$

$$= O(\alpha \text{OPT}).$$

Now we consider the output C^A and w_{out}^A of \mathcal{A} , i.e., for each $y' \in Y'$ the algorithm \mathcal{A} decides which integer weight $w_{out}^A(y')$ of y' is labeled as an outlier. Define $w_{in}^A(y') = w_k(y') - w_{out}^A(y')$. This solution of instance (Y', w') in \mathcal{M}' with $\sum_{y' \in Y'} w_{out}^A(y')$ outliers naturally defines a solution for the original instance X with $|X_{out}| + \sum_{y' \in Y'} w_{out}^A(y')$ outliers: for each center $c \in C^A$ in \mathcal{M}' we define $f(c) \in \mathcal{M}$ as follows: if $c = y^k$ for some y and k , we let $f(c) = y$. Otherwise, $f(c) = y$ for some y and we simply set $f(c) = c$. Now, we choose $f(C^A) := \{f(c) : c \in C^A\} \subseteq Y \subseteq X$ as our set of centers. Moreover, whenever $w_{out}^A(y^k)$ is nonzero, we label $w_{out}^A(y^k)$ arbitrary points in $B_k(y)$ as outliers and similarly, if $w_{out}^A(y)$ is nonzero (and therefore 1), we declare y as an outlier. We call the resulting set of outliers X_{out}^A . Finally, we define all the points in $X_{out} \cup X_{out}^A$ as outliers. Note that we have $\sum_{y' \in Y'} w_{out}^A(y') \leq (1 + \varepsilon)z^A$, so

$$\begin{aligned} |X_{out} \cup X_{out}^A| &\leq |X_{out}| + (1 + \varepsilon)z^A \\ &= |X_{out}| + (1 + \varepsilon)(z - |X_{out}| + \alpha \varepsilon z) \\ &= (1 + O(\alpha \varepsilon))z. \end{aligned}$$

Thus, we only label $(1 + O(\alpha \varepsilon))z$ vertices as outliers, as desired. We now bound the cost of the set $X \setminus (X_{out} \cup X_{out}^A)$ with respect to the set of centers $f(C^A)$.

To that end, we note that for an arbitrary $x \in B_k(y)$ and any $c \in \mathcal{M}'$ we have $\varphi'(y^k, c) = (2^k + d'(y, c))^2$ and therefore

$$\begin{aligned} \varphi(x, f(c)) &\leq (d(x, y) + d(y, f(c)))^2 && \tag{38} \\ &\leq (2^{k+1} + d(y, f(c)))^2 && x \in B_k(y) \\ &\leq 4(2^k + d(y, f(c)))^2 \\ &\leq 4(2^k + d'(y, c))^2 && \text{Definition of } f \text{ and } d' \\ &= 4\varphi'(y^k, c). \end{aligned}$$

Hence, we have

$$\begin{aligned}
 & \varphi(X \setminus (X_{out} \cup X_{out}^A), f(C^A)) && (39) \\
 &= \sum_{y \in Y \setminus X_{out}^A} \varphi(y, f(C^A)) + \sum_{y^k \in Y'} \sum_{x \in B_k(y) \setminus X_{out}^A} \varphi(x, f(C^A)) \\
 &\leq \sum_{y \in Y \setminus X_{out}^A} \varphi'(y, C^A) + 4 \sum_{y^k \in Y'} \sum_{x \in B_k(y) \setminus X_{out}^A} \varphi'(y^k, C^A) && \text{Eq. (38)} \\
 &\leq 4 \sum_{y' \in Y'} w_{in}^A(y') \varphi'(y', C^A) \\
 &\leq 4\beta \cdot 4\varphi'((Y', w_{in}^*), C^*) && \mathcal{A} \text{ is } \beta \text{ apx} \\
 &= O(\alpha\beta\text{OPT}). && \text{Eq. (35)}
 \end{aligned}$$

□

Note that assuming $\frac{\text{OPT}}{2\varepsilon z} \leq \Theta \leq \frac{\text{OPT}}{\varepsilon z}$, [Theorem 13](#) implies that $\sum_j \tau_\Theta(X_j \cap X_{in}^*, Y_j) = O(\text{OPT})$ with positive constant probability and therefore we can assume that $\alpha = O(1)$. This implies that our final solution has a cost of $O(\alpha\beta\text{OPT}) = O(\beta)\text{OPT}$ and moreover the solution outputs at most $(1 + O(\alpha\varepsilon))z = (1 + O(\varepsilon))z$ outliers, as desired. What remains to be discussed is how to remove the assumption that the algorithm knows OPT . To that end, we again run the algorithm for $O(\log(n\Delta))$ guesses of OPT , namely for all values 2^e with $e \in [\log(n\Delta^2)]$ in parallel as before. Each machine will send all $O(\log n)$ respective weighted sets (Y', w) and $|X_{out,j}|$ to the coordinator that outputs the set of candidate centers that minimizes $\varphi'((Y', w_{in}^A), C^A)$ among those where X_{out} satisfies $|X_{out}| \leq z$. [Eq. \(39\)](#) certifies that, with positive constant probability, we get an $O(\beta)$ -approximation of OPT , as needed.