Proceedings of Machine Learning Research vol 132:1–39, 2021

Unexpected Effects of Online no-Substitution k-means Clustering

Michal Moshkovitz MMOSHKOVITZ@ENG.UCSD.EDU
University of California, San Diego

Abstract

Offline k-means clustering was studied extensively, and algorithms with a constant approximation are available. However, online clustering is still uncharted. New factors come into play: the ordering of the dataset and whether the number of points, n, is known in advance or not. Their exact effects are unknown. In this paper we focus on the online setting where the decisions are irreversible: after a point arrives, the algorithm needs to decide whether to take the point as a center or not, and this decision is final. How many centers are needed and sufficient to achieve constant approximation in this setting? We show upper and lower bounds for all the different cases. These bounds are exactly the same up to a constant, thus achieving optimal bounds. For example, for k-means cost with constant $k > 1$ and random order, $\Theta(\log n)$ centers are enough to achieve a constant approximation, while the mere a priori knowledge of n reduces the number of centers to a constant. These bounds hold for any distance function that obeys a triangle-type inequality.

Keywords: unsupervised learning, k-means clustering, online no-substitution, online landscape, identifying principal factors, universality

1. Introduction

Clustering is an unsupervised learning problem where the goal is to group data into a few clusters. It is an important exploratory data analysis step used in various domains like bioinformatics, image analysis, and information retrieval. In the literature, there are many algorithms for clustering in the *offline setting*, where all the points in the dataset are given in advance (Kanungo et al. (2002); Arthur and Vassilvitskii (2007); Aggarwal et al. (2009); Ahmadian et al. (2019)). The output of a center-based clustering algorithm is a set of *centers* in the dataset, where each center is a "representative" of one cluster.

In the *online no-substitution setting*, points in the dataset arrive one after another, and a decision whether to take the current point as a center needs to be made before observing the next point. As a motivating example, Hess and Sabato (2020) suggested a clinical trial of a new drug, where patients are the points and patients given the new drug are the centers. The goal is to provide the new drug to the smallest number of patients, to avoid unnecessary risk (minimize the number of centers), while ensuring a good representation of the entire population in the trial (small k-means cost). Once a patient is out of the clinic, she cannot be tested, and after she took the drug, she cannot undo it — thus, decisions are irreversible. Studying the online setting is more important these days as new data is constantly generated. (L'heureux et al. (2017); Marx (2013)). In the online setting, new factors come into place: the order of the input points (random or worst order) and whether the number of points in the dataset is known in advance or not. It makes sense that the order of the points would impact performance, but by how much? Can a priori knowledge of the size of the dataset improve performance? In this paper we answer these questions and find that these two new factors have unexpected effects on online clustering. Specifically, we show that the ordering of the dataset can exponentially increase the number of centers. We also prove that merely knowing the size of the dataset can reduce the number of centers logarithmically.

1.1. The online framework

To ease the presentation we focus on the k-means cost, though our results apply to more general cost functions, as discussed in Section 5. For any dataset $D = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ and desired number of clusters k, the *k-means cost* is defined as the sum of squared ℓ_2-distances of each point in the dataset to its closest center:

$$cost(c_1, \ldots, c_k) = \sum_{t=1}^{n} \|x_t - c(x_t)\|^2 , \tag{1}$$

where $c(x)$ is the closest center to x, i.e., $c(x) = \arg\min_{c_i \in \{c_1,\ldots,c_k\}} \|x - c_i\|$. We denote by $cost(opt_k)$ the optimal cost using k centers[1]: $cost(opt_k) := \min_{c_1,\ldots,c_k \in D} cost(c_1, \ldots, c_k)$. In the *offline setting* an algorithm receives a dataset D and a desired number of clusters k, and in $poly(n)$ time returns a set of centers $c_1, \ldots, c_\ell \in D$ such that (1) the number of centers, ℓ, is close as possible to k and (2) $cost(c_1, \ldots, c_\ell)$ is close as possible to $cost(opt_k)$.

We focus on the following *online k-means* setting: At each time step, when a new point in the dataset arrives, the algorithm needs to decide whether to take it as a center or not. The decisions cannot be changed after the next point arrives. Points that were not chosen as centers cannot be considered as centers later on, and points that were chosen as centers cannot be removed from the set of centers. This setting was used and motivated in Hess and Sabato (2020).

In this paper, we consider constant approximation algorithms, meaning $cost(c_1, \ldots, c_\ell) \leq a \cdot cost(opt_k)$, where a is some constant. We refer to such a clustering algorithm as a $\Theta(1)$-approximation. The goal of the online algorithm is to minimize the number of centers ℓ and make it as close as possible to k.

1.2. Our contribution

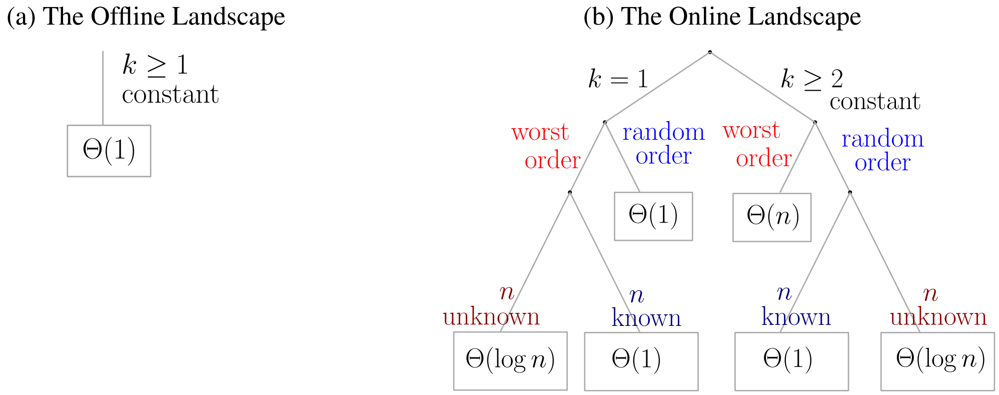
Identifying principal factors. Many factors might affect the quality of online k-means algorithms: the order of the points (random or worst order), dimension size, number of points, and number of clusters. A conceptual contribution of this paper is the observation that the new factors in online clustering (namely, the ordering and whether the number of points is known in advance) significantly influence the optimal algorithms' performance. The dimension, however, plays no role. If the order is arbitrary and $k > 1$, the a priori knowledge of n is irrelevant too.

Entire landscape and optimal bounds. Henceforth we focus on the case of constant k. In the offline setting, an efficient algorithm that returns $\Theta(1)$ centers is known (see Arthur and Vassilvitskii (2007); Aggarwal et al. (2009)). The case of $k = 1$ is more straightforward than that: the optimal center is the average point[2]. That is the end of the story for offline clustering. However, in the online setting, the story only begins. In this paper, we pinpoint the exact number of centers needed and sufficient to achieve a constant approximation for different values of the new factors.

The online landscape that we map is summarized in Figure 1, explained in detail in Sections 3 and 4, and listed next. (i) For $k = 1$: if either the order is random or n is known in advance, then simple algorithms show that $\Theta(1)$ centers are needed and sufficient to achieve a constant approximation. If the order is worst case and n is unknown in advance, $\Theta(\log n)$ centers are needed and sufficient. (ii)

[1] More generally, one can ease the requirement, and allow the centers to be in $\mathbb{R}^d$ and not necessarily in D. This can improve $cost(opt_k)$ only by a factor of 2, see Lemma 10.

[2] If the average point is not in the dataset, take the closest point to it.



(a) The Offline Landscape

(b) The Online Landscape

Figure 1: Comparison between the offline and online settings. All algorithms are $\Theta(1)$-approximation. In rectangles: number of centers (these are optimal) (a) For constant k only constant number of centers are needed in the offline setting (b) In the online setting this paper uncovers a more complex behavior, see the text for details.

For constant $k \geq 2$: if the order is arbitrary, then $\Theta(n)$ centers are needed (and obviously sufficient). If the order is random and n is unknown in advance, then $\Theta(\log n)$ centers are needed and sufficient, but if n is known in advance, then $\Theta(1)$ centers are needed and sufficient.

Universality. Interestingly, the landscape we have described is the same for any cost with a distance function that obeys a triangle-type inequality (e.g., k-medians, or more generally ℓ_p norms with constant p). This is proved in Section 5.

Technical contribution. One of the main technical contributions are two new algorithms for the case that the points arrive in random order and $k \geq 2$. One algorithm is for the case that n is known in advance and thus, the algorithm can observe a small fraction of the data without taking any of these points as centers. Since the order is random, these points represent the entire data, thus it is becoming easier to choose which points to take as centers. The second algorithm is for the case that n is unknown in advance and it uses farthest-first-traversal to find points that should be taken as centers. These algorithms are described in Section 4.

1.3. Related work

Liberty et al. (2016) presented an algorithm for online k-means where centers decisions are irreversible. The order is arbitrary and the cost of a point x is with respect to the closest center in the set of centers selected till x's arrival. Their algorithm adapts the k-means++ algorithm by Arthur and Vassilvitskii (2007) to the online case. Inherently, their algorithm cannot get the optimal bound in the no-substitution setting, as the number of centers depends on the aspect ratio, which can be arbitrarily large. See more details in Appendix A.5. In this paper, we improve both the approximation and the number of centers to the optimal values (see Algorithm 4), assuming the order is random. If the order is arbitrary, then we prove that any approximation algorithm, in the worst case, needs to take almost all points as centers.

A recent work, Hess and Sabato (2020), designed an algorithm that bears some similarity to Algorithm 1. However, Hess and Sabato (2020) considered the statistical question where there is an underlying distribution, as in Ben-David (2007). In this statistical setting the ordering is not a factor. Also, they have to assume that the example space is bounded, and this exclusion of outliers simplifies the solution.

In the streaming model (Aggarwal (2007); Guha et al. (2003); Charikar et al. (2003); Ailon et al. (2009); Shindler et al. (2011); Har-Peled and Mazumdar (2004); Phillips (2016)) points arrive one after another. But, unlike our setting, the algorithm is allowed to choose a center after new points were observed and even go over the points a few times. Braverman et al. (2011); Ackerman and Dasgupta (2014); Raghunathan et al. (2017) assume that the data has some structure, we, however, do not have any assumptions on the data and our algorithms function correctly under any dataset.

In the online facility location, Meyerson (2001), points arrive one at a time, and a set of facilities F is maintained throughout. Each point p incurs instant cost, $d(p, \ell)$, by its closest location $l \in F$. The total cost is $|F| + \sum_p d(p, \ell)$. In our setting, the cost incurs only at the end, but most importantly, we want to minimize the number of centers conditioned on having $O(1)$-approximation. In online facility location, if the distances are too small or too big, then one of the terms, $|F|$ or $\sum_p d(p, \ell)$ can dominate over the other. Several variants of this problem were investigated (e.g., Lang (2018); Feldkord and Meyer auf der Heide (2018)).

2. Preliminaries

In this paper we fix the desired number of clusters to be some constant k. We want to design algorithms that minimize the k-means cost. When the algorithm is understood from the context we denote its cost by $cost(alg)$. We focus on $\Theta(1)$-approximation algorithms, which are formally defined next.

Definition 1 (a-approximation) *We say that a clustering algorithm is an* (a, k)-approximation*,* $a \geq 1$*, for* opt_k *if for every series of* n *data points with probability at least* 0.9

$$\frac{cost(alg)}{cost(opt_k)} \leq a,$$

when k *is understood from the context we simply write* an a-approximation algorithm.

In the paper, we focus on the case that a is some constant, and the goal is to minimize the number of centers. The complementary problem of fixing the number of centers will lead to an infinite approximation in some cases, as our lower bounds suggest. We focus either on a fixed order of examples or random (uniform) order. Note that there are two possible sources of randomness: the algorithm and the points' order. The algorithm should succeed with probability 0.9 (this is some arbitrary constant close to 1) when considering the two sources together.

3. The curious case of $k = 1$

In this section, we focus on the case that there is only one center in the optimal clustering, i.e., $k = 1$. The goal is to find one good enough center. In the offline setting, this problem is trivial, simply take $\frac{1}{n}\sum_{i=1}^{n} x_i$, or a point that is closest to it as the center. So it is surprising that in the online case there is a complex behavior.

It is known that a random point in a cluster is a good enough center of the entire cluster (see Lemma 10 in the appendix). Thus, if the order is random, the algorithm can simply take the first point as a center. If the order is adversarial, but n is known in advance, then a random number in $[n]$ can be taken before the examples were observed. This gives access to a random point, which we know is a good center. For completeness, the proofs of these claims are in Appendix A as Claims 9 and 11.

In case that n is unknown in advance, then $O(\log_c n)$ centers are sufficient to achieve an $O(c)$-approximation, by applying the doubling method, see more details in Claim 12, Appendix A. We prove that for any $c > 1$, any algorithm must take $\Omega(\log_c(n))$ centers for it to be a c-approximation. This means that $\Theta(\log_c(n))$ is tight for any $O(c)$-approximation algorithm.

Theorem 2 *For any integer n and $c \geq 1$, and for any clustering algorithm that is not given n in advance and is a c-approximation, there are n data points and an ordering of them such that the algorithm must take $\Omega(\log_c(n))$ centers with probability at least* 0.8.

We remark that the constant 0.8 is merely a number smaller than 0.9, which appeared in the definition of a c-approximation, Definition 1. One cannot prove that an algorithm must take $\Omega(\log_c(n))$ centers with a probability larger than 0.9 because a valid approximation algorithm can decide with probability 0.1 not to take any center. The idea of the proof is to construct a dataset and an order on them such that the number of centers taken is $\Omega(\log_c n)$ for any c-approximation algorithm. The dataset is composed of $\Omega(\log_c n)$ groups. The groups are evenly spaced on the line (see Figure 2). The number of points in each group is exponential increasing. The points are given, group by group from smallest to largest. Since the algorithm does not know n, the current group can be the last and recall that an approximation algorithm must succeed under any dataset. Thus, the algorithm must take a center from each group. The formal proof is in the appendix.

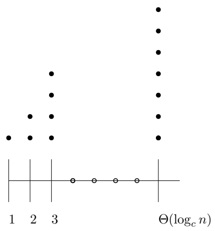


Figure 2: Dataset for proof of Theorem 2

4. The case of constant $k \geq 2$

This section explores the case where the optimal clustering contains k centers, where $k > 1$ is any constant. If the dataset's order can be arbitrary, then any c-approximation deterministic algorithm must take all the points in the dataset. A dataset that shows this is n non-negative points on the line, i.e., $x_t \in \mathbb{R}$, in increasing order where each point is much further than the previous one. Since it is so further away, it has to be taken, otherwise, the rest of the points will be set to 0, which is allowed as points arrive in an arbitrary order. In this case, the largest point has to be taken for the algorithm to be a c-approximation. Thus, all points need to be taken. If the algorithm is stochastic, it needs to take $\Omega(n)$ points as centers. See Liberty et al. (2016) or Claim 13 in Appendix A for more details. As a side note, a follow-up work Bhattacharjee and Moshkovitz (2020), proved that for "structured" data (e.g., points sampled from a k-mixture model), $\texttt{poly}(k \log n)$ centers are enough to achieve $O(k^3)$ approximation. As for the upper bound, an algorithm can take all n data points as centers and achieve a minimal cost of 0. Thus, the upper and lower bounds coincide, up to a constant, when the order is arbitrary.

4.1. Random order and known n

Now let us assume that the data arrives in random order. If n is known in advance, we show an algorithm that takes $\Theta(1)$ centers and is a $\Theta(1)$-approximation, for any constant k. The main idea is to observe a small linear fraction of points without taking any as a center. The option of merely observing data without taking points as centers is possible only because n is known in advance, and the points' order is random. Thus, with a high probability, good enough centers will also be available in the future. Fortunately, as the order is random, the small sample provides enough information.

The algorithm is composed of three phases. In the first phase, it observes a small linear number of points, M_1, without taking any point as a center. It finds an approximately optimal clustering for the points M_1. These centers define a clustering C^{M_1} on the entire dataset, where each point is clustered with its closest center. Since the order is random, these centers are good centers for all large clusters, of size at least $\texttt{poly}(k)$, in the dataset, see Claim 28 in the appendix. Unfortunately, these points cannot be taken as centers in retrospect in our framework. So the algorithm needs to take future points that are close to those centers.

The clustering C^{M_1} represents well only large clusters. To see why, consider, as an extreme example, a small cluster that contains only one point. The algorithm has to take this point as a center when received, or a high cost is incurred. In other words, the algorithm has to take points that are "far". We define "far" by farthest from their center c_i by some threshold t_i. The algorithm needs to decide how to define this threshold, where one option is to take t_i as the max radius of cluster $C_i^{M_1}$. However, this is problematic, as it might cause the algorithm to take too many centers. The problem stems from the fact that we cannot use the sample to define both the centers and t_i's, as the centers are selected to minimize t_i's. To overcome this problem, we introduce an intermediate step, phase 2.

In phase 2, we save another small fraction of points, without taking any of them as center. The distance to the farthest point, $R_{max}[i]$, in this sample, from each center, defines the threshold $t_i = R_{max}[i]$. Since t_i's and the centers are now independent, this will guarantee that the algorithm does not take too many centers that seem far, see Claim 14. In phase 3, we finally take centers. There are two types of centers: (i) points that are close to cluster centers from phase 1, or (ii) points that are considered "far".

To summarize, there are three phases in the algorithm:

- **Phase 1:** The first αn of the points, $\alpha \in (0, 1)$ is a constant to be chosen later, are saved in memory, and the algorithm does not take any of them as centers. We denote this set by M_1. Since n is known, the algorithm can decide not to take αn of the points as centers without increasing the cost by much. After the first αn points arrive the algorithm uses them to find k centers $c_1^{M_1}, \dots, c_k^{M_1}$ that are $\Theta(1)$-approximation clustering for the αn observed points.

- **Phase 2:** The algorithm observes another $\alpha_2 n$ points, M_2, without taking any as center, $\alpha_2 \in (0, 1)$ is another constant to be chosen later. For each center $c_i^{M_1}$ it saves the distance to the farthest point, $R_{max}[i]$, in its cluster among those in M_2.

- **Phase 3:** The algorithm takes the following centers for each center $c_i^{M_1}$: (i) a few close points (ii) points that are farther than the threshold $R_{max}[i]$.

The algorithm's pseudo-code is in Algorithm 1, and its correctness is proved in the next theorem.

Algorithm 1 Online clustering with $k > 1$, n known, random order

```
1:  phase 1: collect data
2:  $M_1$ = save (without taking as center) the first $\lfloor \frac{n}{10^2 k} \rfloor$ points
3:  find offline clustering for $M_1$ with centers $(c_i^{M_1})_{i=1}^k$
4:  phase 2: collect more data to define “far” points
5:  $M_2$ = save (without taking as center) the next $\lfloor \frac{n}{10^5 k^3} \rfloor$ points
6:  for $i = 1$ to $k$ do
7:      $A[i] = \{x \in M_2 : i = \arg\min \|x - c_i^{M_1}\|\}$          # partition $M_2$
8:      $R_{max}[i] = \max_{y \in A[i]} \left\| y - c_i^{M_1} \right\|$          # max distance between $M_2$ and
9:                                                          # current center, if $A[i] = \emptyset$, this is 0
10:     $centers\_counter[i] = 0$                            # init close-centers-counter for phase 3
11: end for
12: phase 3: take centers
13: for the rest of the points $x_t$ (points not received in phase 1 or 2) do
14:     $i^* = \arg\min_i \left\| x_t - c_i^{M_1} \right\|$          # closest center
15:     if $\left\| x_t - c_{i^*}^{M_1} \right\| > R_{max}[i^*]$ then
16:         take $x_t$ as a center                          # points that are far away
17:     end if
18:     if $centers\_counter[i^*] \leq 3k \log(40k)$ then
19:         take $x_t$ as center                            # not enough close points to center $c_i^{M_1}$
20:                                                          # were taken yet $\Rightarrow$ taking a close point
21:         $centers\_counter[i^*]++$
22:     end if
23: end for
```

Theorem 3 *For any constant integer $k \geq 2$, there is an algorithm that given (i) n, the size of the dataset, (ii) k, and (iii) the dataset which appears in a random order, the following holds. With probability at least 0.9, the algorithm takes $\Theta(1)$ centers and $cost(alg) \leq \Theta(1) \cdot cost(opt_k)$.*

To prove the theorem, we need to bound the number of centers the algorithm takes and its approximation. To bound the number of centers, we note that the algorithm takes two types of centers in Line 16 and Line 19. It is easy to bound the second type of centers by $O(k^2 \log k)$. To bound the first type, focus on one cluster $C_i^{M_1}$ in the clustering C^{M_1}. We prove that a significant fraction, a, of the points in $C_i^{M_1}$ are received in phase 2, see Claim 24. We prove that the probability of taking a point as “far” is inversely proportional to $a|C_i^{M_1}|$. The multiplication of the last two terms bounds the expected number of points taken as “far” points. Importantly, this multiplication is a constant. See Claim 14 for more details.

Next, we want to prove that the algorithm is a $\Theta(1)$-approximation, which is formally proved in Claim 15 in the appendix. We show that for each optimal clustering, the algorithm takes as center a point that is a good enough center for the entire cluster. For that aim, we separate the analysis into two cases depending on the size of the cluster: small (of size smaller than $\texttt{poly}(k)$) or large. We start with the small-size analysis. Focus on a small cluster C_i^*. Take a point in $x_0 \in C_i^*$ that is a good center for all the points in C_i^*. As a side note, since the cluster is small, there might be only one point in C_i^*. This point, most likely, will not be received in the first two phases. Suppose that x_0 is in a cluster $C_i^{M_1}$ with center $c_i^{M_1}$. Let us focus on all points A in $C_i^{M_1}$ that are farther than x_0 from the center. It A is small, then none of the points in A are chosen in phase 2, see Claim 25, and x_0 will be taken as a center. If A is large, then C_i^* can be merged into a different optimal cluster. This is formally proved in Claim 16.

Moving on to the case of large optimal cluster C_i^*, from Section 3, we know that most points $Good_i \subseteq C_i^*$ in the cluster can be a good enough center for the entire cluster. This implies that the fraction of points we get from $Good_i$ in phases 1 and 2 is between $(\alpha + \alpha_2)/2$ and $2(\alpha + \alpha_2)$ as the order is random, see Claim 24. Focus on the cluster $C_i^{M_1}$ with the center $c_i^{M_1}$ containing most of the remaining points from $Good_i$. There are two cases: either $C_i^{M_1}$ includes mostly points from $Good_i$ and then the algorithm probably takes a point from $Good_i$, or this cluster C_i^* can be merged into a different optimal cluster. This is formally proved in Claim 17.

Before moving to the following case, a few remarks. The paper does not try to optimize the dependence on k, where the number of centers is $\texttt{poly}(k)$, and the approximation is $\exp(k \log k)$. Indeed, in a follow-up work Hess et al. (2021), a new algorithm was presented with improved dependency on k. Second, the work Indyk (1999) designed a sublinear time algorithm for k-medians, which has some similarities to Algorithm 1. One major difference is the algorithm's treatment of far points. While they can consider the furthest points in the entire dataset as far points, we need to decide online if a point is far or not. For that, we had to introduce phase 2. A detailed discussion of more differences can be found in Appendix A.4.

4.2. Random order and unknown n

In the last section, we designed an algorithm that uses $\Theta(1)$ centers and achieves $\Theta(1)$-approximation, when k is a constant, if the number of points, n, is known in advance. In contrast, in this section, we show that if n is unknown in advance, any algorithm must take $\Omega\left(k \log \frac{n}{k}\right)$ centers. The lower-bound dataset is similar to the $\Omega(n)$ lower bound used in the worst-case order, where points are in $\mathbb{R}$ with increasing distances. The idea is that an approximation algorithm must take the $k-1$ largest points as each step; otherwise, the data stream can stop. In the rest of the section, we design a new algorithm that achieves a matching upper bound, up to a constant.

Theorem 4 *For any scalar $c > 1$, integers $k \geq 2$ and n, and for any clustering algorithm that does not know what n is and is a c-approximation, there are n points that arrive uniformly at random and the algorithm must take $\Omega\left(k \log \frac{n}{k}\right)$ centers with probability at least 0.7.*

4.2.1. THE CASE OF $k = 2$

To simplify the presentation, we start with the case that $k = 2$. Many of the ideas are also applicable to the case of $k > 2$. We prove that $\Theta(\log n)$ centers are needed and sufficient for a $\Theta(1)$-approximation, when n is unknown and $k = 2$. We show a simple algorithm that saves only a small number of bits in memory (more specifically, it saves the first example and only one more number), achieves $\Theta(1)$-approximation, and takes at most $\log(n) + 2$ centers. Our results are tight when n is unknown, as we show a matching lower bound.

Theorem 5 *There is an online algorithm such that if the examples are received with random order (n does not have to be known) then with probability at least 0.9 it holds that number of centers is $O(\log n)$ and $cost(alg) \leq \Theta(1) \cdot cost(opt_2)$.*

Algorithm 2 Online clustering with $k = 2$, n unknown, random order

```
1: take $x_1$ as a center
2: $x := x_1$ (save first data point)
3: max_dis := 0
4: for $t = 2, \dots$ do
5:     if $\|x_t - x\| >$ max_dis then
6:         take $x_t$ as a center
7:         max_dis := $\|x_t - x\|$
8:     end if
9: end for
```

To bound the number of centers taken by Algorithm 2, note that the i-th example is chosen as a center only if it is the furthest from x (recall that x is the first example). This will happen with probability $\frac{1}{i-1}$. Thus, the expected number of centers is the n-th harmonic number, which is about $\log n$. To prove the algorithm is a $\Theta(1)$-approximation, in a high level, we separate the analysis into two cases: either the two clusters are close to each other or not. It the two clusters are close, we can treat them as one cluster with center x. Using Lemma 10, x is a good center. If the two clusters are far apart, then, most probably, the first point from the second cluster is furthest away from x among all points received so far.

Formalizing the last argument, we want to show that the algorithm takes two points as centers that are good representatives of each of the optimal clusters C_1^*, C_2^*. Denote by $Good_i$, $i = 1, 2$, the set of points in each optimal cluster C_i^* that can be taken as a center without increasing the cluster's cost by much. From the same arguments as in Section 3, we know that $Good_i$ is a significant fraction of C_i^*. With high enough probability, the first point from each cluster $i = 1, 2$ is a good center, i.e., in $Good_i$. Specifically, x, the first point, is a good center for its cluster C_1^*. Thus the algorithm needs to take as a center one point in $Good_2$ (the algorithm might take many more points as centers to achieve this goal). We hope to show that the algorithm takes the first point from C_2^* as a center. Denote by $y_2^* \in Good_2$ the closest point in $Good_2$ to x. Focus on the set B of points that will interfere in taking the first point in $Good_2$ as a center

$$B = \{y_1 \in C_1^* : \|y_1 - x\| \geq \|y_2^* - x\|\}.$$

There are two cases: either B is small compared to C_2^* or not. If it is small, then most likely, the first point from $C_2^* \cup B$ is in C_2^*, or in different words, the first point from C_2^* will arrive before any point in B. Thus the first point from C_2^* will be taken as center. In the other case, B is large compared to C_2^*. This means merging C_1^* and C_2^* together increases the cost by only a constant factor. Thus, x can be a good center for C_2^* too.

4.2.2. THE CASE OF CONSTANT $k > 2$

For the more general case of constant $k > 2$ we present Algorithm 4 that uses $O\left(k \log \frac{n}{k}\right)$ centers, which matches the lower bound of Theorem 4, and is a $\Theta(1)$-approximation for constant k.

We want to borrow the main idea of Algorithm 2: if clusters are far apart, take the first point from each optimal cluster, otherwise merge clusters. Algorithm 2 detects the arrival of the first point x_t from a new cluster by measuring the distance to the first point received x, see Line 5. This technique will not work for $k > 2$, as the next example demonstrates. Focus on $k = 3$ and three well-separated

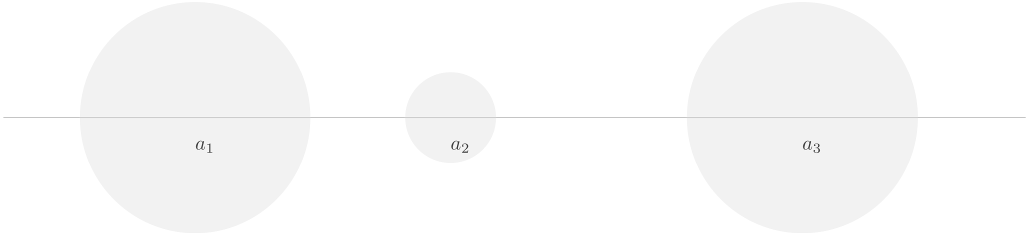


Figure 3: Motivation for Algorithm 4: detecting the middle cluster.

clusters on a line with centers $a_1 \ll a_2 \ll a_3$, where the middle cluster is much smaller in size than the other two clusters, see Figure 3. Most likely, the first points will be from the first and third clusters. It is unclear how to detect the arrival of the first point from the second cluster.

Inspired by Figure 3, taking k points that are farthest from each other, the first point from a new cluster is one of those k points. The main idea of Algorithm 2 is to use the known farthest-first-traversal algorithm as a subroutine. Perhaps surprisingly, This subroutine is beneficial for a different cost function, k-center Dasgupta (2013). For completeness, the farthest-first-traversal algorithm appears as Algorithm 3.

Algorithm 3 Farthest-first-traversal(M, s, k)

```
$S = \{s\}$
for $t = 2, \ldots, k$ do
    $v = \arg\max_{x \in M-S} \min_{y \in S} \|x - y\|^2$
    $S = S \cup \{v\}$
end for
return $S$
```

The farthest-first-traversal algorithm returns k (a parameter) points that are far away from each other in a given dataset M. Specifically, it starts with some point $s \in M$ that is given as an input. Then it takes the point $x_2 \in M$ that it furthest away from s. Then a point $x_3 \in M$ that maximizes the distance to S, where the distance is equal to $dis(S, x) = \min_{y \in S} \|y - x\|$.

For our purposes, the primary claim we need from farthest-first-traversal is that if S was returned, then the distance of any point x to S is smaller than the distance between any two points inside S, as the next lemma proves. For completeness, the proof is in Appendix C.

Lemma 6 *Suppose S = Farthest-first-traversal(M, s, k) and $x \in M - S$, then*

$$\min_{y \in S} \|x - y\| \leq \min_{y_1, y_2 \in S} \|y_1 - y_2\|.$$

The online clustering algorithm, Algorithm 4, saves in memory all the points encountered so far. Deciding whether to take the current point as a center or not uses the farthest-first-traversal algorithm that picks k far away points. The current point is taken as a center if chosen as one of the k points.

Algorithm 4 Online clustering with $k \geq 2$, n unknown, random order

```
Take $x_1, \ldots, x_k$ as centers and save them in $M$
for $t = k+1, \ldots$ do
    $M = append(M, x_t)$
    $S =$ Farthest-first-traversal$(M, x_1, k)$
    if $x_t \in S$ then
        take $x_t$ as a center
    end if
end for
```

Theorem 7 *There is an online algorithm such that for any constant k, if the examples are received with random order (n does not have to be known) then with probability at least 0.9 the algorithm uses $O(\log n)$ centers and achieves $\Theta(1)$-approximation.*

The number of centers Algorithm 4 takes is bounded because the i-th example, $i > k$, is chosen as a center only if it is one of the k points defining the unique farthest-first-traversal for the first i points. This will happen with probability $\frac{k}{i}$. Thus, the expected number of centers can be calculated using the n-th harmonic number, which is about $k \log \frac{n}{k}$.

Next, we show that the algorithm is a $\Theta(1)$-approximation. We remark that the paper does not try to optimize the dependence on k, where the approximation is $\exp(k \log k)$. Focus on an optimal cluster C_i^* and its first point x_i. It can be the case that there is another optimal cluster C_r^* and a point $x_r \in C_r^*$ that causes x_i not to be taken as a center. This situation can happen, but we want to show that it occurs with a small probability. To do so, we show that the number of points that might interfere with taking x_i as a center is small. We need to define the set of points that interfere. Intuitively, these are points that have a considerable distance compared to points in C_i^*. One option to define this considerable distance is the distance of C_i^*'s closest point to other clusters. Another is the furthest. It turns out that both of these options will not work. The furthest will not work because it is not necessarily a good center to C_i^*. The closest will not work because there might be many points in other clusters that are farthest from this point. We need to define this distance somewhere between the farthest and the closest. Lemma 6 implies that if the first point from a new cluster is not taken as a center, then there are many points in a different cluster C_r^* with cost higher than merging C_i^* to a different optimal cluster. See the proof of Claim 20 for more details.

5. General cost function

So far we focused on the k-means cost, but one can consider a more general cost:

$$cost(c_1, \ldots, c_k) = \sum_{x \in D} d(x, c(x)), \tag{2}$$

where d is a distance function[3] and $c(x) = \arg\min_{c_i \in \{c_1, \ldots, c_k\}} d(x, c_i)$. Specifically, we focus on distance function that satisfy a version of a triangle inequality: there is a *constant* $D \geq 1$ such that

$$\forall u, v, w. \quad d(u, v) \leq D \cdot (d(u, w) + d(w, v)). \tag{3}$$

[3] d should be (i) non negative ($d(x, y) \geq 0$) (ii) $d(x, x) = 0$ and (iii) symmetric $d(x, y) = d(y, x)$ (iv) satisfy triangle inequality

For the k-means cost $d(x, y) = \|x - y\|^2$ and indeed Claim 34 proves that Inequality 3 holds with $D = 2$. In the k-medians case $d(x, y) = \|x - y\|_1$ and Inequality 3 holds with $D = 1$.

We show that our results, which are summarized in Figure 1, hold for any cost function that satisfies Inequality 3. In the rest of this section we outline some of the proof ideas. First observe that the triangle inequality immediately implies a similar claim as Lemma 10; see Appendix C for the proof:

Lemma 8 *For any $x_1, \ldots, x_n, \mu$, and integer j chosen uniformly at random from $[n]$, it holds that*

$$\mathbb{E}_{j \in [n]} \left[\sum_{i=1}^{n} d(x_i, x_j) \right] \leq 2D \cdot \sum_{i=1}^{n} d(x_i, \mu).$$

This implies that a random point in a cluster is a good enough center for the entire cluster.

The curious case of $k = 1$. In case that (i) the order is random or (ii) the order is worst case and n is known, Lemma 8 establishes that a constant number of centers is sufficient to achieve a constant approximation. In case that the order is worst case and n is unknown (i) the doubling technique (see Algorithm 5) takes $O(\log n)$ points as centers and achieves $O(cD) = O(c)$ approximation, see Claim 22 (ii) assuming there are n points $x_1, \ldots, x_n$ with $d(x_i, x_j) = |j - i|$ (e.g., $x_i = i$), we can show a lower bound of $\Omega(\log n)$ centers, see Theorem 21.

The case of constant $k \geq 2$. The algorithms presented in this paper, Algorithms 1 and 4 function correctly under a general cost function when the norm is replaced with $d(\cdot, \cdot)$. Indeed, the proofs of Theorems 3 and 7 use the general cost (as defined in Equation 2). To avoid the specificity of the k-means cost, these proofs use Lemma 8 heavily. To prove the lower bounds presented in this paper we need, for any $c > 0$, a series of points $x_1, \ldots, x_n$ such that $d(x_{i+1}, x_i) \geq c \cdot d(x_i, x_{i-1})$ (it is easy to find such a series in $\mathbb{R}$). Given this series, the proof for the general case is the same as the proofs of Claims 13 and 18.

6. Conclusion

In this paper, we showed optimal bounds for online clustering when the number of centers, k, is a constant, i.e., we showed matching upper and lower bounds. We uncovered a complex behavior in the online setting compared to the offline setting. Specifically, in the former, new factors arise: the order of the dataset and knowing in advance the size of the dataset. These factors have dramatic effects on online algorithms as illustrated in Figure 1. These bounds hold for any cost function that obeys triangle-type inequality. In the paper, we designed new algorithms that can learn under different circumstances. Specifically, if the order is random we designed algorithms that take as centers only $\Theta(1)$ points if n is known, and $O(\log(n))$ centers if n is unknown, both are optimal bounds. These algorithms work without any assumptions on the data.

Acknowledgments

I thank Sanjoy Dasgupta for introducing me to the fascinating world of online k-means, reviewing initial drafts of this paper, and for many stimulating discussions.

References

Margareta Ackerman and Sanjoy Dasgupta. Incremental clustering: The case for extra clusters. In *Advances in Neural Information Processing Systems*, pages 307–315, 2014.

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.

Charu C Aggarwal. *Data streams: models and algorithms*, volume 31. Springer Science & Business Media, 2007.

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*, 2019.

Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *Advances in neural information processing systems*, pages 10–18, 2009.

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.

Shai Ben-David. A framework for statistical clustering with constant time approximation algorithms for k-median and k-means clustering. *Machine Learning*, 66(2-3):243–257, 2007.

Robi Bhattacharjee and Michal Moshkovitz. No-substitution k-means clustering with adversarial order. *arXiv preprint arXiv:2012.14512*, 2020.

Vladimir Braverman, Adam Meyerson, Rafail Ostrovsky, Alan Roytman, Michael Shindler, and Brian Tagiku. Streaming k-means on well-clusterable data. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 26–40. Society for Industrial and Applied Mathematics, 2011.

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

Sanjoy Dasgupta. Geometric algorithms lecture notes, 2013. URL https://cseweb.ucsd.edu/~dasgupta/291-geom/kcenter.pdf.

Björn Feldkord and Friedhelm Meyer auf der Heide. Online facility location with mobile facilities. In *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures*, pages 373–381, 2018.

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.

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

Tom Hess and Sivan Sabato. Sequential no-substitution k-median-clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 962–972. PMLR, 2020.

Tom Hess, Michal Moshkovitz, and Sivan Sabato. A constant approximation algorithm for sequential no-substitution k-median clustering under a random arrival order. *arXiv preprint arXiv:2102.04050*, 2021.

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. ACM, 1999.

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. In *Proceedings of the eighteenth annual symposium on Computational geometry*, pages 10–18, 2002.

Harry Lang. Online facility location against at-bounded adversary. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1002–1014. SIAM, 2018.

Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. An algorithm for online k-means clustering. In *2016 Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 81–89. SIAM, 2016.

Alexandra L'heureux, Katarina Grolinger, Hany F Elyamany, and Miriam AM Capretz. Machine learning with big data: Challenges and approaches. *IEEE Access*, 5:7776–7797, 2017.

Vivien Marx. Biology: The big challenges of big data. *Nature*, 498:255–260, 2013.

Adam Meyerson. Online facility location. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 426–431. IEEE, 2001.

Jeff M Phillips. Coresets and sketches. *arXiv preprint arXiv:1601.00617*, 2016.

Aditi Raghunathan, Prateek Jain, and Ravishankar Krishnawamy. Learning mixture of gaussians with streaming data. In *Advances in Neural Information Processing Systems*, pages 6605–6614, 2017.

Michael Shindler, Alex Wong, and Adam W Meyerson. Fast and accurate k-means for large datasets. In *Advances in neural information processing systems*, pages 2375–2383, 2011.

This appendix includes the proofs of the theorems and claims appearing throughout the paper. It consists of three sections: the first proves the main text's claims, the second technical claims on random samples, and the third general stand-alone technical claims.

Appendix A. Proofs of main theorems and claims

A.1. Random order, $k = 1$

If the points' order is random, then there is a simple algorithm that uses only one center while preserving a constant approximation: simply taking the first point.

Claim 9 *If the data points appear in a random order, there is an online algorithm that uses only one center and with probability at least* 0.9 *it holds that* $cost(alg) \leq 20 \cdot cost(opt_1)$.

The main tool in proving the theorem, which will also be useful in cases where $k > 1$, is the following known lemma that states that a random point in a cluster can be the good enough center for this cluster.

Lemma 10 *Let* $x_1, \ldots, x_n \in \mathbb{R}^d$ *it holds that*

$$\mathbb{E}_{j\in[n]}\left[\sum_{i=1}^{n} \|x_i - x_j\|^2\right] = 2\sum_{i=1}^{n} \|x_i - \mu\|^2 \,,$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ *is the optimal center and* j *is chosen uniformly at random from* $[n]$.

For completeness, the proof of the lemma appears in Appendix C. To prove Lemma 9 we use Lemma 10 and Markov's inequality.

A.2. Arbitrary order, $k = 1$

We proceed to the case where the order of the data points is not random but it can appear in the worst order possible. In this case we witness a surprising result — it matters whether n is known or not. If n is known in advance, then the algorithm can take one random point as a center and it will be a constant approximation algorithm.

Claim 11 *There is an online algorithm that receives as input* n*, the size of the dataset and* n *data points such that the following holds. For any order of the data points, the algorithm uses only one center and with probability at least* 0.9 *it holds that* $cost(alg) \leq 20 \cdot cost(opt_1)$.

The correctness of the algorithm is similar to Claim 9. If n is unknown in advance, one can use the doubling method.

Claim 12 *For any* $c > 1$ *there is an algorithm that obtains* $O(c)$*-approximation with* $O(\log_c n)$ *centers, no matter what the order is and even if* n *is unknown.*

Intuitively, since the algorithm does not know the value of n, it guess it and applies the algorithm from Theorem 11. The algorithm starts by assuming that n is small ($n = 1$). Once more data is arrived, it increases the value of n to c, and then to c^2 and so on. For each value of n, it applies the algorithm from Theorem 11, i.e., picks one random point in the next n data points and take only this point as a center. This algorithm uses only $O(\log_c n)$ centers. Intuitively, it is $O(c)$-approximation because in the one before the last iteration most of points are still not received and the algorithm chooses a random point among many of them which yields a good center by Lemma 10. The pseudocode appears in Algorithm 5.

Algorithm 5 Online clustering with $k = 1$, n unknown, arbitrary order

```
$last = 0$, $n' = 1$, $i^* = 1$
for $t = 1, \dots$ do
    if $t == last + i^*$ then
        take $x_t$ as a center
    end if
    if $t == last + n'$ then
        pick $i^* \in [n']$ at random
        $last = last + n'$, $n' := c \cdot n'$
    end if
end for
```

In the first line of Algorithm 5, it initialize the parameters. $last$ is the number of points we encountered before the current round, n' is our current guess of the number of points, and i^* is the point the algorithm will choose as center in the current round. In the loop, after the algorithm encounters n' points, it increases n' by a factor of c, we signal that we have encountered more examples by increasing $last$ and a new point, i^* to be our next center is picked.

Proof [of Claim 12] It is easy to bound the number of centers Algorithm 5 uses, it's bounded by the number of times n' is increased plus 1. The latter is bounded by $O(\log_c(n))$.

Next we prove that the algorithm is an $O(c)$-approximation. For ease of notation, iteration 0 is the first iteration and in this iteration $n' = c^0 = 1$. The last iteration i^* is the one where $\sum_{i=0}^{i^*} c^i > n$ for the first time. This means that

$$\frac{c^{i^*} - 1}{c - 1} \le n < \frac{c^{i^*+1} - 1}{c - 1}. \tag{4}$$

We now focus on iteration $i^* - 1$ where the value of n' is c^{i^*-1}. From Equation 4, we know that $n \le \frac{n' \cdot c^2 - 1}{c-1} \Rightarrow \frac{n(c-1)+1}{c^2} \le n' \Rightarrow \frac{n}{2c} \le n'$ (in the last equation we used that w.l.o.g $c \ge 2$). In words, in the one before the last iteration n' is big compared to n, it's at least a fraction $1/2c$ of n.

From Lemma 10 and Markov's inequality we get that for at most $\frac{n}{20 \cdot c}$ of the data points x it holds that $cost(x) > 40 \cdot c \cdot cost(opt_1)$. Thus with probability at least 0.9 the algorithm chooses a data point x out of the n' points at iteration $i^* - 1$ with $cost(x) \le 40c \cdot cost(opt_1)$. ■

Proof [of Theorem 2] The examples are $1, 2, 3, \dots$ and so on. Example with value i appears $(7c)^i$ times (except the last value that might appear less than that). The points are given, group by group from smallest to largest. Note that there are $\Omega(\log_c n)$ different points. We will show that any c-approximation algorithm must take $\Omega(\log_c n)$ examples as centers with probability at least 0.8. This will finish the proof.

There are two cases: either for each example (out of the $\Omega(\log_c n)$ different examples) the probability it will be taken as a center (at least one of the $(7c)^{t^*}$ instances of this example) is at least 0.9 or not. First case implies, using Claim 33, that with probability at least 0.8 the algorithm takes as centers at least half of the examples, i.e., it takes $\Omega(\log_c n)$ examples as centers. Second case implies that there is an example t^* such that probability that the algorithm will not take as center is more than 0.1. We will show that the algorithm is not a c-approximation and we will reach a contradiction.

We focus on the series $1, \dots, t^*$. Note that since n is unknown the algorithm works the same up until t^*, no matter if there are more examples after t^* or not. Also recall that the algorithm should work correctly for any dataset. This means that also for the shorter series that includes only examples $1, \dots, t^*$, the probability that t^* is taken as a center is smaller than 0.1. We will prove that if t^* is not taken, which happens with probability more than 0.1, it holds that $c \cdot cost(opt_1) < cost(alg)$ for the dataset that includes the points in $1, \dots, t^*$, which is a contradiction to the assumption that the algorithm is a c-approximation (i.e., that $cost(alg) \leq c \cdot cost(opt_1)$ with probability at least 0.9).

The cost of the algorithm is at least $cost(alg) \geq (7c)^{t^*}$, as there are $(7c)^{t^*}$ examples with value t^* and the distance to the closest center is at least 1. Let's compare it to the cost of the optimal solution. One can take example x_{t^*} and the optimal cost is only smaller, i.e., $cost(opt_1) \leq \sum_{i=1}^{t^*} (7c)^i (t^* - i)^2$. Using Claim 32 we have that

$$cost(opt_1) \leq 6(7c)^{t^*-1}$$

This implies that

$$c \cdot cost(opt_1) \leq 6c(7c)^{t^*-1} < (7c)^{t^*} \leq cost(alg),$$

which is a contradiction to the c-approximation of the algorithm. ■

A.3. Arbitrary order, $k \geq 2$

If the order of the dataset can be arbitrary, then any c-approximation algorithm must basically take all the points in the dataset. The idea of the proof is the following. We present n non-negative points on the line, i.e., $x_t \in \mathbb{R}$, in increasing order. Each point is much further than the previous one. Since it is so further away it has to be taken, otherwise, the rest of the points will be set to 0, which is allowed as points arrive in an arbitrary order. In this case the largest point has to be taken or the ratio $\frac{cost(alg)}{cost(opt_2)}$ is arbitrarily large. Thus, all the points need to be taken if the algorithm is deterministic. If the algorithm is stochastic $\Omega(n)$ of the points need to be taken with probability at least 0.8.

Any algorithm can take n data points as centers and achieve the minimal cost of 0. Thus, we conclude that the upper and lower bounds coincide, up to a constant, when the order is arbitrary.

Claim 13 *For any integers $k \geq 2$ and n, any scalar $c > 1$, and for any clustering algorithm that is a c-approximation (even if n is known) there are n points and an ordering of them such that the algorithm must take $\Omega(n)$ centers with probability at least* 0.8.

Proof We will define a series of points in $\mathbb{R}$, $0 = x_1 < x_2 < \ldots < x_n$ such that any c-approximation algorithm must take $\Omega(n)$ points as centers with probability at least 0.8. This will finish the proof.

If the probability that each example is taken as center is at least 0.9, then with probability at least 0.8 the algorithm takes $0.5n = \Omega(n)$ points as centers, see Claim 33, and the theorem follows. Focus on the first example x_{t^*} such that the probability that the algorithm will not take it as a center is more than 0.1. We will show that the algorithm is not a c-approximation. We focus on the series of points $x_1, \dots, x_{t^*}, 0, 0, \dots, 0$. Recall that for the algorithm to be a c-approximation, it should function correctly under any dataset and specifically one the latter. Note that up until example x_{t^*} the two series ($(x_1, x_2, \dots, x_n)$ and $(x_1, \dots, x_{t^*}, 0, 0, \dots, 0)$) are the same, thus the probability that x_{t^*} will not be taken as a center is more than 0.1.

One can take x_{t^*} and 0 as centers, the optimal cost can only be smaller, thus

$$\mathit{cost}(\mathit{opt}_k) \leq \sum_{t=2}^{t^*-1} \|x_t - x_1\|^2 = \sum_{t=2}^{t^*-1} x_t^2.$$

Since the algorithm did not take x_{t^*} as a center we have that

$$\mathit{cost}(\mathit{alg}) \geq \|x_{t^*} - x_{t^*-1}\|^2 .$$

For this lower bound to work, we take the series of examples such the following strict inequality holds

$$\mathit{cost}(\mathit{alg}) \geq (x_{t^*} - x_{t^*-1})^2 > c \cdot \sum_{t=2}^{t^*-1} x_t^2 \geq c \cdot \mathit{cost}(\mathit{opt}_k).$$

In different words,

$$x_{t^*} > x_{t^*-1} + \sqrt{c \cdot \sum_{t=2}^{t^*-1} x_t^2}$$

And we are done since with probability more than 0.1 it holds that $\mathit{cost}(\mathit{alg}) > c \cdot \mathit{cost}(\mathit{opt}_k)$, i.e., the algorithm is not a c-approximation. ■

A.4. Random order, $k \geq 2$, known n

To make our claims more general, we consider an arbitrary cost that its distance function satisfy a version of a triangle inequality: there is a *constant* D such that

$$\forall u, v, w. \quad d(u, v) \leq D \cdot (d(u, w) + d(w, v)). \tag{5}$$

We call such a cost a *D-cost.*

Proof [of Theorem 3] We prove the claim under the general cost function, see Section 5. For the k-means cost $d(x, y) = \|x - y\|^2$, Inequality 3 holds with $D = 2$. To ease the analysis, we add an initial step to the algorithm and take the first point as a center. This will increase the number of centers by only 1.

A few notation first. Number of examples is n, number of examples the algorithm saves in the first phase is equal to $|M_1| = \alpha n$, where $\alpha := \frac{\lfloor \frac{0.01}{k} n \rfloor}{n}$, and $|M_2| = \lfloor \alpha_2 n \rfloor$ is the number of points received in phase 2, where $\alpha_2 = \alpha/10^5 k^3$.

In the proof we consider three clusterings of the entire dataset. The first, $C^{M_1} = (C_i^{M_1})_{i=1}^k$, is the optimal clustering with respect to the points M_1 the algorithm saves in the first phase. The second is the optimal clustering $C^* = (C_i^*)_{i=1}^k$ induced by the entire dataset. And the third $C = (C_i)_{i=1}$ is induced by all the centers taken by the algorithm in the third phase. We prove that with probability at least 0.9 the following two claims hold for any constant k

1. The number of centers the algorithm takes is $\Theta(1)$
2. $cost(C) \leq \Theta(1) \cdot cost(C^*)$.

This is proved in Claims 14 and 15. ■

Claim 14 *Algorithm 1 takes as center at most $O(k^5)$ points, with probability at least* 0.98.

Proof The algorithm takes as a center two types of points. Either close (Line 19 in Algorithm 1) or far (Line 16). Bounding the number of close points by $O(k^2 \log k)$ is easy, as it follows immediately from the definition of the algorithm — the algorithm only takes $O(k \log k)$ close points per cluster. The interesting claim is bounding the number of centers that are far.

The key idea is that the points in the second phase, M_2, that constitute a good representation of the clustering created after the first phase, C^{M_1}, in the sense that the algorithm receive a linear number of points, $\beta|C_i|$, from each large enough cluster $C_i^{M_1} \in C^{M_1}$ (this is true using Claim 24). A point $x \in C_i^{M_1}$ is considered *far* from the cluster center, $c_i \in C_i^{M_1}$, if x is furthest from all the $\beta|C_i^{M_1}|$ points received in the second phase. The probability that x will be far is very small, only ${}^1\!/\!{}_{(\beta|C_i^{M_1}|)}$. Thus, the total number of centers the algorithm takes is only a constant. For small clusters, the algorithm can take all the points in the cluster, and still not take too many centers.

More formally, for each cluster $C_i^{M_1} \in C^{M_1}$ we separate the analysis depending on whether the points left from the cluster in is small or large. If it's small, i.e., $|C_i^{M_1} - M_1| \leq \frac{400k}{\alpha_2}$, then the algorithm takes at most all the points remaining in this cluster, $\frac{400k}{\alpha_2} = O(k^4)$. Since there are at most k small clusters (because there are k clusters in total), the total number of centers taken because of small clusters is only $O(k^5)$.

For large clusters, $|C_i^{M_1} - M_1| > \frac{400k}{\alpha_2}$, Claim 24 and union bound prove that, with probability at least 0.99, for all large clusters, many points are received in the second phase, i.e.,

$$\frac{\alpha_2}{2}|C_i^{M_1} - M_1| \leq |(C_i^{M_1} - M_1) \cap M_2|. \tag{6}$$

A point $x \in C_i^{M_1}$ is taken as a far point if it's far from the center of the cluster $c_i^{M_1}$ compared to all points in this cluster received in phase 2, i.e, if $d(x, c_i^{M_1}) > d(y, c_i^{M_1})$ for all $y \in C_i^{M_1}$ received in phase 2. From Inequality 6, the probability of taking a point as far, happens with probability at most

$$\frac{2}{\alpha_2|C_i^{M_1} - M_1|}.$$

We can also deduce from Inequality 6 that there are at most $(1 - \frac{\alpha_2}{2})|C_i^{M_1} - M_1|$ points in $C_i^{M_1}$ that are received in the third phase. So in total, the expected number of points for cluster $C_i^{M_1}$ taken as far points is bounded by

$$\frac{2}{\alpha_2|C_i^{M_1} - M_1|} \cdot \left(1 - \frac{\alpha_2}{2}\right)|C_i^{M_1} - M_1| = \frac{2 - \alpha_2}{\alpha_2}.$$

Use Markov's inequality to show that with probability at least 0.99 all the large clusters cause at most $O(\frac{2-\alpha_2}{\alpha_2} \cdot k^2) = O(k^5)$ far centers. Summing the two cases, the number of points taken as centers by the algorithm is $O(k^5)$ with probability at least 0.98. ■

Claim 15 *The cost of Algorithm 1 is bounded by* $\Theta(1) \cdot cost(opt_k)$, *with probability at least* 0.95.

Proof Denote by C the clustering returned by Algorithm 1. We want to prove that $cost(C)$ is at most some function of k times $cost(opt_k)$. We will prove a stronger result. We will prove that when the algorithm is given a dataset and a parameter k it performs well compared to any $opt_{k'}$ for any integer $k' \leq k$. Namely, we prove that for any integer $k' \leq k$ with probability at least $1 - \frac{5k'}{100k}$

$$cost(C) \leq \left(\frac{26kD^5a}{\alpha}\right)^{k'} \cdot cost(opt_{k'})$$

Fix k. We prove this claim by induction on k'. For $k' = 1$, recall that to simplify the analysis the algorithm takes the first point as center, and thus the claim follows immediately, similarly to Claim 9.

For $k' > 1$, we use Claims 16 and 17 with $\delta = \frac{5}{100k}$. Specifically, if there is a cluster such that $cost(opt_{k'-1}) \leq \frac{26kD^5a}{\alpha} cost(opt_{k'})$ (it's case 2 in both of the claims) then using the induction hypothesis

$$cost(C) \leq \left(\frac{26kD^5a}{\alpha}\right)^{k'-1} \cdot cost(opt_{k'-1}) \leq \left(\frac{10^7D^5k^5}{\alpha^2}\right)^{k'} cost(opt_{k'}).$$

Otherwise, these claims prove that

$$cost(C) \leq 5D^2k \cdot cost(opt_{k'}) \leq \left(\frac{10^7D^5k^5}{\alpha^2}\right)^{k'} cost(opt_{k'}).$$

■

In the following claim we show that for a small optimal cluster (smaller than $poly(k)$) either a good center will be taken for it or it can be merged into another cluster.

Claim 16 *Fix* $\delta \in (0,1)$ *and a D-cost. Suppose Algorithm 1 has the following parameters.*

- *First phase: use* αn *points and an* a*-approximation algorithm with* $k \geq k'$ *clusters*
- *Second phase: use* βn *points with* $\beta \leq \frac{\alpha\delta^2}{32k}$ *and* $\alpha + \beta \leq \frac{\delta}{2}$

For any optimal clusters C_i^*, $i \in [k']$, *such that* $|C_i^*| < \frac{16}{\alpha\delta}$, *with probability at least* $1 - \delta$ *one of the following holds:*

1. *a point* $v \in C_i^*$ *will be taken as a center with* $\sum_{x \in C_i^*} d(x, v) \leq 2D \cdot cost(opt_{k'})$
2. $cost(opt_{k'-1}) \leq \frac{13kD^5a}{\alpha} \cdot cost(opt_{k'})$

Proof Focus on the closest point $x_i \in C_i^*$ to the center c_i^*. The cost of taking x_i as a center to C_i^* is

$$\sum_{x \in C_i^*} d(x, x_i) \leq D \sum_{x \in C_i^*} d(x, c_i^*) + D \sum_{x \in C_i^*} d(c_i^*, x_i) \leq 2D \sum_{x \in C_i^*} d(x, c_i^*), \tag{7}$$

in the first inequality we used Inequality 3 and in the second we used the fact that x_i is closest to c_i^* than all points in C_i^*.

The probability that x_i will be received in phase 1 or 2 is $\alpha+\beta \leq \delta/2$. The rest of the analysis will focus on the case that x_i was received only on phase 3. The point x_i is in some cluster $C_i^{M_1} \in C^{M_1}$ with center $c_i^{M_1}$. Focus on all points in $C_i^{M_1}$ that are furthest from the center $c_i^{M_1}$ than x_i :

$$A = \left\{ x \in C_i^{M_1} \wedge x \neq x_i : d(x, c_i^{M_1}) \geq d(x_i, c_i^{M_1}) \right\}.$$

We separate the analysis into two cases depending on the the size of A. If $|A| \leq \frac{\delta}{2\beta}$. Then, using Claim 25, we know that with probability at least $1 - \frac{\delta}{2}$ the algorithm will not get a member of $|A|$ in the second phase, and thus a x_i will be taken as center at phase 3, which completes the proof of case 1 in the claim.

If $|A| > \frac{\delta}{2\beta} \geq \frac{16k}{\alpha\delta}$, then there is an optimal cluster C_j^* with center c_j^* that includes at least $\frac{16}{\alpha\delta}$ of the points in A. We will show that C_i^* can be merged into C_j^*. We want to bound the cost of the following clustering with $k'-1$ centers: $c_1^*, \ldots, c_{k'}^*$ without c_i^* and all points in C_i^* will be assigned to c_j^*. The cost is equal to

$$\sum_{x \in C_i^*} d(x, c_j^*) + \sum_{r \neq i} \sum_{x \in C_r^*} d(x, c_r^*)$$

Let us bound the first sum using Inequality 2 twice

$$\begin{aligned} \sum_{x \in C_i^*} d(x, c_j^*) &\leq D \sum_{x \in C_i^*} d(x, x_i) + D^2 \sum_{x \in C_i^*} d(x_i, c_i^{M_1}) + D^2 \sum_{x \in C_i^*} d(c_i^{M_1}, c_j^*) \\ &\leq 2D^2 \sum_{x \in C_i^*} d(x, c_i^*) + D^2 \sum_{x \in A \cap C_j^*} d(x, c_i^{M_1}) + D^2 |A \cap C_j^*| d(c_i^{M_1}, c_j^*), \end{aligned}$$

where in the first inequality: the first term is bounded by Inequality 7; the second term by the fact that for each member $x \in A$ have $d(x, c_i^{M_1}) \geq d(x_i, c_i^{M_1})$ and $|A \cap C_j^*| \geq \frac{16}{\alpha\delta} \geq |C_i^*|$; for the third term we used again $|A \cap C_j^*| \geq |C_i^*|$.

To bound the last expression we use Claim 29 and Claim 30 to deduce that with probability at least $1-\delta$, $cost(opt_{k'-1})$ is bounded by

$$2D^2 cost(opt_{k'}) + \frac{5kD^4 a}{\alpha} cost(opt_{k'}) + \frac{6kD^5 a}{\alpha} cost(opt_{k'}) \leq \frac{13kD^5 a}{\alpha} cost(opt_{k'}).$$

■

In the following claim we show that for a large optimal cluster, either a good center will be taken for it or it can be merged into another cluster.

Claim 17 *Fix $\delta \in (0,1)$ and D-cost. Suppose Algorithm 1 has the following parameters.*

- *First phase: use αn points and an a-approximation algorithm with $k \geq k'$ clusters*
- *Second phase: use βn points with $\alpha + \beta \leq \frac{1}{4k}$.*
- *Third phase: at least $3k \log \frac{2}{\delta}$ are taken as centers as close points, for each cluster*

For any optimal clusters C_i^*, $i \in [k']$, such that $|C_i^*| \geq \frac{16}{\alpha\delta}$, with probability at least $1-\delta$ one of the following holds:

1. *a point $v \in C_i^*$ will be taken as a center with that $\sum_{x \in C_i^*} d(x,v) \leq 5D^2 \cdot cost(opt_{k'})$*
2. *$cost(opt_{k'-1}) \leq \frac{26kD^5a}{\alpha} \cdot cost(opt_{k'})$*

Proof Denote by G_i the $\frac{|C_i^*|}{2}$ closest points in C_i^* to the center c_i^*. The cost of taking any $g \in G_i$ as a center to C_i^* is small

$$\sum_{x \in C_i^*} d(x,g) \leq D \sum_{x \in C_i^*} d(x,c_i^*) + D \sum_{x \in C_i^*} d(g,c_i^*) \leq 5D^2 \sum_{x \in C_i^*} d(x,c_i^*), \tag{8}$$

where in the first inequality we used Inequality 3 and in the second we used the definition of G_i, Lemma 8, and Markov's inequality. We will show that either the algorithm takes a point from G_i as a center in the third phase or C_i^* can be merged into another optimal clustering without harming the cost by much.

There is a cluster in $C_i^{M_1} \in C^{M_1}$ that contains at least $|G_i|/k$ of the points in C_i^*. There are two cases: either $C_i^{M_1}$ contains at least $|G_i|/k$ points from a different optimal cluster C_j^* or not. If it contains that many points from a different cluster C_j^*, then, by Claim 31, with probability at least $1 - \frac{\delta}{2}$ it holds that

$$cost(opt_{k'-1}) \leq \frac{26k^2D^5a}{\alpha} \cdot cost(opt_{k'}),$$

which completes the proof of case 2 in the claim. Otherwise, from Claim 24, with probability at least $1 - \delta/2$ at most $2(\alpha+\beta)|G_i|$ points from G_i are received in phase 1 and 2. Or, in different words at least $(\frac{1}{k} - 2\alpha - 2\beta)|G_i| \geq \frac{1}{2k}|G_i|$ points in $G_i \cap C_i^{M_1}$ are received in phase 3.

In the current case, $C_i^{M_1}$ contains at most $|G_i|$ points from clusters different from C_i^*. Thus, from Corollary 27 the first $3k \log \frac{2}{\delta}$ points in $C_i^{M_1}$ in third phase with probability $1 - \delta/2$ is a member of G_i and this completes case 1 in the claim. ■

Comparison between Algorithm 1 and Indyk (1999): The work Indyk (1999) designed a sublinear time algorithm for k-medians, which is similar to the k-means problem discussed in this paper. Their algorithm has some similarities to Algorithm 1, that works in the case that number of points in the dataset, n, is known in advance and the order of the dataset is random. Several differences cause the analysis and algorithm to be different:

1. Parameter regime: "large" cluster in inherently different in the two algorithms. In Indyk (1999), large means $O(\sqrt{n})$, as they cannot take more points for the algorithm to be with sublinear time. On the other hand, for Algorithm 1 "large" means some constant fraction because the algorithm is allowed to take only a constant number of centers in the second phase.

2. Centers from phase 1: Indyk (1999) simply takes the centers that were chosen in phase 1. In our framework this is not allowed since once a center was observed the algorithm cannot retake it. To overcome this obstacle we take centers that are close to the centers chosen in phase 1. But then we need to decide what is close which complicates the analysis.

3. Far points: in Indyk (1999), far points are the furthest points from the cluster defined in phase 1. We cannot use this definition in our framework. To resolve this issue, we add an intermediate step where the algorithm saves a constant fraction of number of points to set a bar that defines far. In the last phase, only points that are above the bar, are taken as centers. Note that the points in the first phase cannot be used to define the bar, as they were used to define the cluster.

A.5. Random order, $k \geq 2$, unknown n

A.5.1. LOWER BOUNDS

Claim 18 *For any integer n, any scalar $c > 1$, and for any clustering algorithm that does not know what n is and is a c-approximation, there are n points such that the algorithm must take $\Omega(\log n)$ centers with probability at least 0.7.*

To prove the theorem we take the same dataset as in the proof of Claim 13. In this construction, at each iteration the point with the maximal value has to be taken, otherwise the examples can stop and the algorithm will not be a c-approximation. If the order is random, there will be $\Omega(\log n)$ points that are maximal, as the probability that the i-th point to be maximal is about $1/i$. Thus $\Omega(\log n)$ have to be taken as centers.

Proof [of Claim 18] We will use the same series of points in $\mathbb{R}$ that was used in the proof of Theorem 13. We say that a point in the i-th iteration is *maximal* if it's the largest value so far. We prove the following two claims:

1. For any c-approximation it must take at least 0.5 of the maximal points with probability at least 0.8.
2. With probability at least 0.99 there are $\Omega(\log n)$ maximal points.

Once we prove the two steps we are done.

Claim 1 - There are two cases (i) for each point if it's a maximal point, the probability the algorithm takes it as a center is at least 0.9 (ii) there is a point x that if x is a maximal point the probability the algorithm takes it as a center is less than 0.9. In case (i), using Claim 33, we know that with probability at least 0.8 the algorithm takes half of the maximal points. In case (ii), from the same argument as the proof of Theorem 13 we have a contradiction to the assumption that the algorithm is a c-approximation using the dataset $1, \ldots, x$.

Claim 2 - denote by X the random variable that is equal to the number of maximal points and denote by X_i the binary random variable that is equal to 1 if the i-th example is maximal, and otherwise $X_i = 0$. For any i, $\mathbb{E}[X_i]$ is equal to the probability that the i-th example is the largest than all previous examples. This probability is equal to $\mathbb{E}[X_i] = 1/i$. Thus

$$\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \frac{1}{i} \leq \log n + 1.$$

Note that X_i's are independent and thus we can use Hoeffding's inequality. ■

For the case of random order, unknown n and general k, we will show a lower bound of $\Omega(k \log \frac{n}{k})$ on the number of centers and a matching upper bound.

Proof [of Theorem 4] The high-level idea is similar to the one in Claim 18. We will have a series of numbers $w_1 < \ldots < w_n$ with increasing distances $w_{i+1} - w_i$. We will show that since n is unknown, the $k-1$ largest number at each step have to be taken as centers for the algorithm to be a c-approximation algorithm, and there are $\Omega\left(k \log \frac{n}{k}\right)$ points that are at some step among the $(k-1)$ largest numbers.

We say that the point received in the t-th iteration is *k-maximal* if it is one of the $k-1$ largest values among the points received till iteration t. We choose our dataset such that at each iteration the

$k-1$ largest points so far must be taken as centers. Namely, the $i+1$ point in the dataset is chosen such that

$$(w_{i+1}-w_i)^2 > c\sum_{j=1}^{i}(w_i-w_j)^2.$$

Suppose an algorithm does not take a k-maximal point at time t, and denote by w_i the k-th largest point at this time step. The algorithm's cost, $cost(alg)$, is at least $(w_{i+1}-w_i)^2$. On the other hand, one can take the k largest points as centers and get a bound on the optimal cost $cost(opt) \leq \sum_{j=1}^{i}(w_i-w_j)^2$. By the choice of the dataset we get that if a k-maximal point is not taken as a center, than $cost(alg) > c \cdot cost(opt)$.

Expected number of points that are k-maximal is

$$k-2+\sum_{j=k-1}^{n}\frac{k-1}{j} = \Omega\left(k\log\frac{n}{k}\right).$$

The theorem follows from the following two claims:

1. For any c-approximation it must take at least 0.5 of the k-maximal points with probability at least 0.8.
2. With probability at least 0.99 there are $\Omega(k\log\frac{n}{k})$ maximal points.

The two claims follow from similar arguments as the proof of Claim 18. ■

A.5.2. UPPER BOUNDS

Proof [of Theorem 5] We start by bounding the number of expected centers the algorithm uses. Denote by X the random variable that is equal to the number of centers chosen. Denote by X_i the random variable that is 1 if the i-th point is taken as center and 0 otherwise. Then, the expected number of centers the algorithm chooses is equal to

$$\mathbb{E}[X] = \mathbb{E}[\sum_{i=1}^{n}X_i] = \sum_{i=1}^{n}\mathbb{E}[X_i] = \sum_{i=1}^{n}\Pr(i\text{-th point is a center})$$

By the definition of the algorithm X_1 is always 1. For $1 < i \leq n$ it holds that $X_i = 1$ only if the i-th point is the furthest away from the first point among points $1,\dots,i-1$. The probability that the i-th point is the furthest is equal to $\frac{1}{i-1}$. Thus

$$\mathbb{E}[X] = 1+\sum_{i=2}^{n}\frac{1}{i-1} = 1+\sum_{i=1}^{n-1}\frac{1}{i} \leq \log(n-1)+2.$$

From Markov's inequality with probability at least 0.99 number of centers is $O(\log n)$.

We are now left with proving that Algorithm 2 is a $\Theta(1)$-approximation. Focus on the optimal clustering. Denote by C_1^* the points in the first cluster and by C_2^* the points in the second cluster. We define the set of *good* points for a cluster r ($r=1,2$) as the set of points that taking them as a center will not increase the cost by much. More formally,

$$Good_r = \{y_r \in C_r^* | \sum_{x_i\in C_r^*}\|x_i-y_r\|^2 \leq 100\sum_{x_i\in C_r^*}\|x_i-\mu^*(r)\|^2\},$$

where $\mu^*(r) = \frac{1}{|C_r^*|}\sum_{x_i \in C_r^*} x_i$ is the optimal center for cluster C_r^*. From Lemma 10 and Markov's inequality we know that $Good_r$ is large. Specifically,

$$\frac{|Good_r|}{|C_r^*|} \geq 1 - \frac{1}{50}.$$

Thus, using union bound, with probability at least $1 - \frac{2}{50}$ the first point the algorithm encounters from each cluster r is good.

Fix the first point the algorithm encounters by x, w.l.o.g $x \in C_1^*$. Define by $y_2^* \in Good_2$ the closest point to x in $Good_2$, i.e.,

$$y_2^* = \arg\min_{y_2 \in Good_2} \|y_2 - x\| .$$

Denote by B all the points in C_1^* that are further from x than y_2^*, i.e.,

$$B = \{y_1 \in C_1^* : \|y_1 - x\| \geq \|y_2^* - x\|\}.$$

There are two cases

1. $|B| \leq 0.01|C_2^*|$: we will show that most probably, the first point the algorithm encounters in C_2^* will be chosen as a center. We know that

   $$|B| \leq 0.01|C_2^*| \leq 0.02|Good_2|.$$

   Thus, with probability at least $1 - 0.02$, the first point in $Good_2 \cup B$ is in $Good_2$ and the algorithm takes it as a center.

2. $|B| > 0.01|C_2^*|$: we will show that C_1^* and C_2^* can be viewed as one cluster with x as its center without harming the cost by much.

$$\begin{aligned}
\sum_{y \in C_1^* \cup C_2^*} \|y - x\|^2 &= \sum_{y_1 \in C_1^*} \|y_1 - x\|^2 + \sum_{y_2 \in C_2^*} \|y_2 - x\|^2 \\
&= \sum_{y_1 \in C_1^*} \|y_1 - x\|^2 + \sum_{y_2 \in C_2^*} \|(y_2^* - x) + (y_2 - y_2^*)\|^2 \\
&\leq \sum_{y_1 \in C_1^*} \|y_1 - x\|^2 + 2|C_2^*| \, \|y_2^* - x\|^2 + 2 \sum_{y_2 \in C_2^*} \|y_2 - y_2^*\|^2 \\
&\leq \sum_{y_1 \in C_1^*} \|y_1 - x\|^2 + 2 \cdot 100 \sum_{y_1 \in B} \|y_1 - x\|^2 + 2 \sum_{y_2 \in C_2^*} \|y_2 - y_2^*\|^2 \\
&\leq 201 \sum_{y_1 \in C_1^*} \|y_1 - x\|^2 + 2 \sum_{y_2 \in C_2^*} \|y_2 - y_2^*\|^2 \\
&\leq 20100 \cdot cost(opt_2),
\end{aligned}$$

where the first inequality follows from Claim 34, the second from the definition of B, and the third from the definition of $Good$. ■

Proof of Theorem 7

Proof We prove that for any constant k with probability at least 0.9

1. Number of centers is bounded by $O(k \log \frac{n}{k})$
2. $cost(alg) \leq \Theta(1) \cdot cost(opt_k)$

These claims are proved in Claims 19 and 20. We prove the claim under the general cost function, see Section 5. ■

Claim 19 *Algorithm 4 takes as center at most $O(k \log \frac{n}{k})$, with probability at least* 0.99.

Proof To bound the number of centers the algorithm uses, we use a similar argument as in Theorem 5. The algorithm takes the first k points. For $i > k$ we want to find the probability that the i-th point is selected as a center. This happens if among the i points read so far, this point is one of the k members in the Farthest-first-traversal. The probability for this is

$$\frac{k}{i}.$$

Thus the expected number of centers taken by the algorithm throughout its entire run is

$$k + \sum_{j=k+1}^{n} \frac{k}{j} = k + k \left(\sum_{i=1}^{n} \frac{1}{i} - \sum_{i=1}^{k} \frac{1}{i} \right) = \Theta\left(k \log \frac{n}{k}\right).$$

Use Markov's inequality to prove that with probability 0.99 number of centers takes by the algorithm is at most $O\left(k \log \frac{n}{k}\right)$. ■

Claim 20 *The cost of Algorithm 4 is bounded by $\Theta(1) \cdot cost(opt_k)$, with probability at least* 0.96.

Proof We will show that Algorithm 4 is a $\Theta(1)$-approximation. To achieve that we will prove something stronger: when running the algorithm with parameter k, then for any $1 \leq k' \leq k$ with probability at least $1 - \frac{4k'}{100k}$,

$$cost(alg) \leq (10^5 D^3)^{k'} k^{2k'} \cdot cost(opt_{k'}).$$

Fix k. We prove this claim by induction on k'. For $k' = 1$ the claim follows immediately, similarly to Algorithm 5, as we always take the first point as center.

Denote the k' optimal clusters by $C_1^*, \ldots, C_{k'}^*$. Focus on a cluster C_i^*. The idea of the proof is that either the first point in C_i^*, most probably, is chosen as a center or the entire cluster can be added to another cluster. When the first point $x \in C_i^*$ arrives, its closest point is $y \notin C_i^*$. If x is not chosen as a center, then there are k centers from $k-1$ clusters that the distance between any two is larger than $d(x, y)$. There are two cases, as in Theorem 5, either this is a common scenario, and then C_i^* can be added to another cluster, or it's rear case and this means that with high probability x will be chosen as a center.

We define the set of *good* points for a cluster $r \in [k]$ as the set of points that taking them as a center will not increase the cost by much. More formally,

$$Good_r = \left\{ y_r \in C^*_r | \sum_{x_i \in C^*_r} d(x_i, y_r) \leq 100Dk \sum_{x_i \in C^*_r} d(x_i, \mu^*(r)) \right\},$$

where $\mu^*(r)$ is the optimal center for C^*_r. From Lemma 10 and Markov's inequality we know that $Good_r$ is large. Specifically,

$$\frac{|Good_r|}{|C^*_r|} \geq 1 - \frac{1}{50k}.$$

Thus, using union bound, we know that with probability at least $1 - \frac{1}{50}$ the first point the algorithm encounters from each cluster r is in in $Good_r$.

Fix a cluster C^*_i for some $i \in [k]$. The distance between $Good_i$ and a point y is defined as

$$dis(Good_i, y) = \min_{y' \in Good_i} d(y', y).$$

Denote by N the set of points that are not in the cluster C^*_i (i.e., in $C^* - C^*_i$) and are one of the ${}^{|C^*_i|}/_{100k}$ closest points to $Good_i$. Denote the max distance in N by dis, i.e., the distance that is ${}^{|C^*_i|}/_{100k}$-closest to $Good_i$:

$$dis = \max_{y' \in N} \min_{y \in Good_i} d(y', y).$$

Denote the point the achieves this minimum in $Good_i$ by x_i. We want to define *bad* points that can cause the algorithm not to take the first point from C^*_i as a center. For that we first take for each cluster $r \neq i$ an arbitrary point in $x_r \in Good_r$. Now we are ready to define the bad points:

$$B = \cup_{r \neq i} \left\{ y \in C^*_r : d(y, x_r) \geq \frac{dis}{2D} \right\}.$$

There are two cases: either B is large compared to C^*_i, or it is not.

If $|B| \geq \frac{1}{100k}|C^*_i|$: we will show that taking the $k-1$ centers — all the centers x_r's without x_i — is a good enough clustering and then the claim follows by the induction assumption. To prove that, first take, among all points in N, one $y_j \in N$ that it's closest to its representative x_j:

$$y_j = \arg\min_{\substack{j \in [k] - \{i\} \\ y \in N \cap C^*_j}} d(y, x_j).$$

In particular, since $y_j \in N$ and the definitions of x_i and dis we get that

$$d(x_i, y_j) \leq dis.$$

By the definition of y_j we also have that for every $r \in [k] - \{i\}$ and $y_r \in B \cap C^*_r$,

$$d(y_j, x_j) \leq d(y_r, x_r).$$

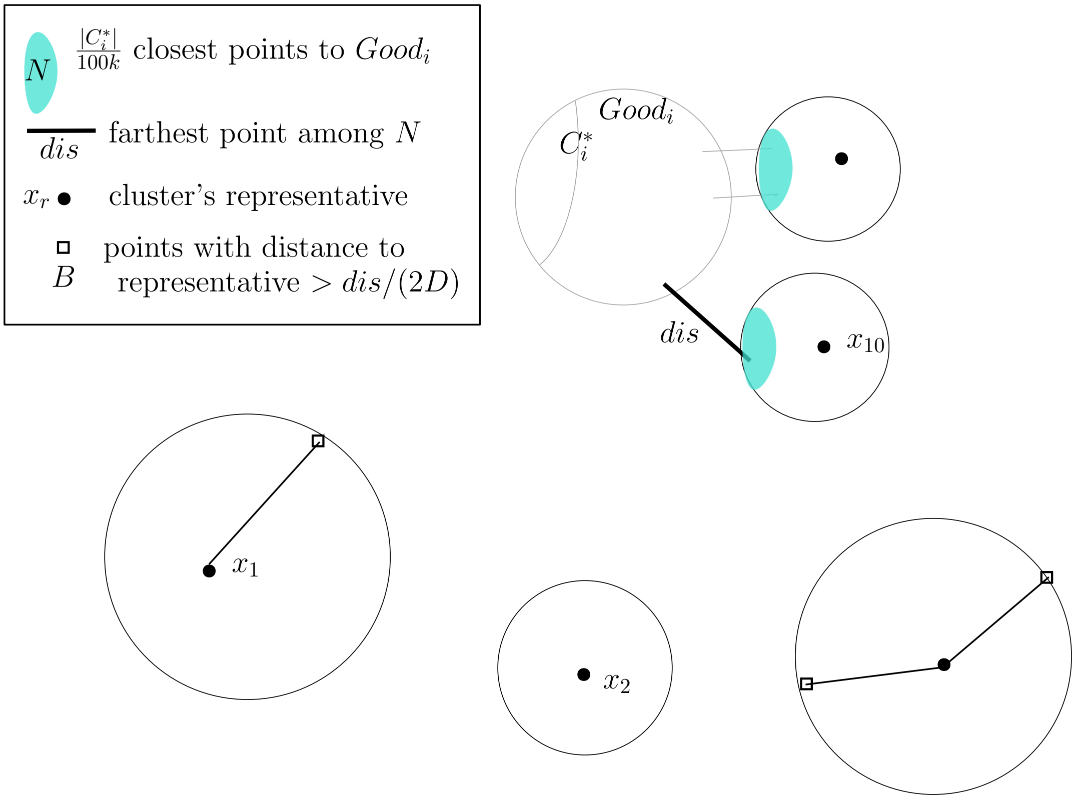


Figure 4: Notations used in the proof of Theorem 7

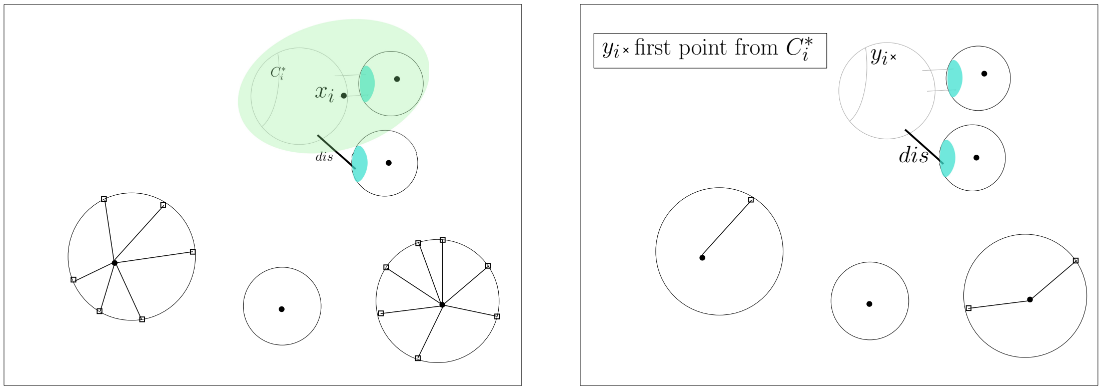


Figure 5: Bounding the cost for a cluster C_i^*. (a) if many points are in B merge C_i^* into its closest cluster (b) otherwise the first point in C_i^* will be taken

We can use these two observations to bound $cost(opt_{k'-1})$ in terms of $cost(opt_{k'})$:

$$\begin{aligned}
cost(opt_{k'-1}) &\leq \sum_{y\in C_i^*} d(y,x_j) + \sum_{r\neq i}\sum_{y\in C_r^*} d(y,x_r)\\
&\leq D\sum_{y\in C_i^*} d(y,x_i) + D^2\sum_{y\in C_i^*} d(x_i,y_j) + D^2\sum_{y\in C_i^*} d(y_j,x_j) + \sum_{r\neq i}\sum_{y\in C_r^*} d(y,x_r)\\
&\leq D^2\sum_{y\in C_i^*} d(x_i,y_j) + D^2\sum_{y\in C_i^*} d(y_j,x_j) + D\sum_{r}\sum_{y\in C_r^*} d(y,x_r)\\
&= D^2|C_i^*|d(x_i,y_j) + D^2|C_i^*|d(y_j,x_j) + D\sum_{r}\sum_{y\in C_r^*} d(y,x_r)\\
&\leq D^2|C_i^*|dis + D^2|C_i^*|d(y_j,x_j) + D\sum_{r}\sum_{y\in C_r^*} d(y,x_r)\\
&\leq 100D^2k|B|dis + 100D^2k\frac{|C_i^*|}{100k}d(y_j,x_j) + D\sum_{r}\sum_{y\in C_r^*} d(y,x_r)\\
&\leq 201D^2k\sum_{r}\sum_{y\in C_r^*} d(y,x_r) \leq 10^5D^3k^2\cdot cost(opt_{k'})
\end{aligned}$$

where the second inequality follows from Claim 34 (or Inequality 3 for the general cost), the forth since $y_j \in N$, the sixth because y_j is closest to its representative from all N, and the last inequality follows from the definition of good.

From the induction assumption we know that

$$\begin{aligned}
cost(alg) &\leq (10^5D^3)^{k'-1}(k-1)^{2k'}\cdot cost(opt_{k'-1})\\
&\leq (10^5D^3)^{k'-1}k^{2k'}\cdot(10^5D^3)^2\cdot cost(opt_{k'})\\
&= (10^5D^3)^{k'}k^{2k'}\cdot cost(opt_{k'})
\end{aligned}$$

And this proves the claim in this case. Let's move on to the next case.

If $|B| < \frac{1}{100k}|C_i^*|$: We will show that with probability at least $1-\frac{4}{100k}$, the first point from C_i^* will be chosen as a center and is in $Good_i$. Then, if there is no cluster C_i^* with $|B| \geq |C_i^*|/(100k)$

we can use union bound over all the k' centers will finish the proof. Focus on the time where the first point in C_i^*, $y_i \in C_i^*$ was given. With probability at least $1 - 1/50k$ it is in $Good_i$. With probability at least $1 - \frac{1}{100k}$ a point from $|B|$ was not chosen yet (as B is much smaller than C_i^*). With probability at least $1 - \frac{1}{100k}$ a point from N was not chosen yet (as N is much smaller than C_i^*), thus minimal distance from a point in $Good_i$ to another is at least dis.

For the sake of contradiction, let us assume that y_i, the first point from C_i^*, was not chosen as a center. Then all points S returned by the farthest-first-traversal algorithm must came from at most $k-1$ optimal clusters. From the Pigeonhole principle, there are two points $y_1, y_2 \in S$ that are in the same cluster C_j^*. From Lemma 6 we have that

$$dis \leq d(y_1, y_2).$$

From Claim 34 (or Inequality 3) we have that $d(y_1, y_2) \leq D \cdot d(y_1, x_j) + D \cdot d(x_j, y_2)$. This implies that $d(y_1, x_j) \geq dis/(2D)$ (or $d(y_2, x_j) \geq dis/(2D)$). This means that $y_1 \in B$. Which is a contradiction to the assumption that no point in B was chosen yet.

■

Comparison between Algorithm 4 and Liberty et al. (2016): Liberty et al. (2016) design an online clustering adaptation to the k-means++ algorithm Arthur and Vassilvitskii (2007). In the k-means++ algorithm a point is chosen with probability $^{\text{distance}}/_Z$, where Z is the normalization factor. In the online case Z is unknown as it depends on future points and thus cannot be calculated. Instead, Liberty et al. (2016) suggests to start with some small Z and to keep increasing it by a factor of 2. This implies that the number of centers taken by the algorithm depends on the scale of the data. The scale of data D is summarized in the aspect ratio parameter $\gamma = \frac{\max_{v,v' \in D} \|v - v'\|}{\min_{v,v' \in D} \|v - v'\|}$. Liberty et al. (2016) takes $O(\log n(\log \gamma + \log n))$ centers and achieves $O(\log n)$-approximation. Inherently, this algorithm depends on γ as it must reach the scale of the data to be able to take only a small subset of points as centers. Because of the dependence on γ, their algorithm cannot achieve the optimal bound. In this paper, we improve both the quality of the approximation and the number of centers by the algorithm to the optimal values in case the order is random (see Algorithm 4).

A.6. General cost function

Claim 21 *Assume there are n points with $d(x_i, x_j) = |j - i|$. For any clustering algorithm that is not given the dataset size in advance and is a c-approximation compared to opt_1, there are n data points and an ordering of them such that the algorithm must take $\Omega(\log_c(n))$ centers with probability at least 0.8.*

Proof The proof is the same as in Theorem 2, where now the examples are $x_1, \ldots, x_n$ instead of $1, \ldots, n$. ■

Claim 22 *For any $c > 1$ there is an algorithm that obtains $O(cD)$-approximation with $O(\log_c n)$ centers, no matter what the order is and even if n is unknown.*

Proof Similarly to the proof of Theorem 12, in the one before the last iteration, n' is big compared to n, it's at least a fraction $1/2c$ of n. From Lemma 8 and Markov's inequality we get that for at most $\frac{n}{20 \cdot c}$ of the data points x it holds that $cost(x) > 40 \cdot D \cdot c \cdot cost(opt_1)$. Thus with probability at least 0.9 the algorithm chooses a data point x out of the n' points at the one before the last iteration with $cost(x) \leq 400cD \cdot cost(opt_1)$. ■

Appendix B. Random order

In this section we present and prove general claims in case data appears in random order. These claims will be helpful to analyze Algorithm 1. In Section B.1 we analyze the expected appearance of a predetermined set in a random sample. In Section B.2 we connect the optimal clustering and a clustering based on a random sample.

B.1. Random sample

The next claim shows that for any predetermined set, with a high probability the sample is a good representation of this set. This probability depends on the set and sample sizes; if they are bigger the probability is higher.

Claim 23 *Fix a dataset D, a subset $S \subseteq D$ and a scalar $0 \leq \beta \leq 1$. Assume a subset $S' \subseteq D$ with $\frac{|S'|}{|D|} = \beta$ is chosen uniformly at random. Then, for any $a > 0$,*

$$\Pr\left(\left||S \cap S'| - |S|\frac{|S'|}{|D|}\right| \geq \sqrt{\frac{|S'||S|}{a|D|}}\right) \leq a$$

Proof Denote by X the random variable that is equal to $|S \cap S'|$. Order the members in S in some arbitrary order. Denote by X_i, $i = 1, \ldots, |S|$, the binary variable that is equal to 1 if the i'th member in S is also in S', and $X_i = 0$ otherwise. Note two basic properties of the random variables X_i's:

$$\mathbb{E}[X_i] = \beta, \tag{9}$$

and for every $i \neq j$ it holds that

$$\mathbb{E}[X_i X_j] = \Pr(X_i = 1 \wedge X_j = 1) = \frac{|S'|(|S'| - 1)(|D| - 2)!}{|D|!} = \frac{|S'|(|S'| - 1)}{|D|(|D| - 1)} \leq \beta^2, \tag{10}$$

where the second equality holds since we can view the process of taking S' as if we order all the points in D and then take the first $|S'|$ members; with this view, the ith member has $|S'|$ places and the jth member has $|S'| - 1$ places and then arrange all the other members, $(|D| - 2)!$ options.

We now analyze the expectation and the variance of X.

$$\mathbb{E}[X] = \sum_{i=1}^{|S|} \mathbb{E}[X_i] = \beta|S|.$$

Bounding the variance of X

$$\begin{aligned}
Var[X] &= \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{|S|} X_i\right)^2\right] - \left(\sum_{i=1}^{|S|} \mathbb{E}[X_i]\right)^2 \\
&= \mathbb{E}\left[\sum_{i \neq j} X_i X_j\right] + \mathbb{E}\left[\sum_{i=1}^{|S|} X_i^2\right] - \sum_{i,j} \mathbb{E}[X_i]\,\mathbb{E}[X_j] \\
&= \sum_{i \neq j} \mathbb{E}\left[X_i X_j\right] + \sum_{i=1}^{|S|} \mathbb{E}\left[X_i\right] - \sum_{i,j} \mathbb{E}[X_i]\,\mathbb{E}[X_j] \\
&\leq \beta^2 |S|^2 + \beta|S| - \beta^2|S|^2 = \beta|S|
\end{aligned}$$

where the fourth equality follows from the fact that $X_i^2 = X_i$ as X_i is a binary random variable and the inequality follows from Equation 9 and Equation 10. Next we use Chebyshev's inequality which is the following bound for any $C > 0$:

$$\Pr(|X - \mathbb{E}[X]| \geq C) \leq \frac{Var[X]}{C^2}.$$

In our case, take

$$C = \sqrt{\frac{Var[X]}{a}} \leq \sqrt{\frac{\beta|S|}{a}}$$

and Chebyshev's inequality implies that

$$\Pr\left(|X - \beta|S|| \geq \sqrt{\frac{\beta|S|}{a}}\right) \leq a.$$

■

The next claim shows that for any large predetermined set, the sample is a good representation of this set. More specifically, for any large enough fixed set S, its proportion, $|S\cap S'|/|S'|$, in the sample S', is roughly its proportion, $|S|/|D|$, in the dataset D. Equivalently, the proportion of points, $|S\cap S'|/|S|$, in S that are in the sample, is about $\beta|S|$ where $\beta = |S'|/|D|$ is the fraction of sample of the entire dataset.

Claim 24 *Fix a dataset D, a subset $S \subseteq D$ and a scalar $0 \leq \beta \leq 1$. Assume a subset $S' \subseteq D$ of size $|S'| = \beta|D|$ is chosen uniformly at random. For any $a > 0$, if $|S| \geq \frac{4}{\beta a}$ then*

$$\Pr\left(\frac{\beta}{2}|S| \leq |S \cap S'| \leq 2\beta|S|\right) \geq 1 - a$$

Proof From Claim 23 we know that

$$\Pr\left(||S \cap S'| - \beta|S|| \geq \sqrt{\frac{\beta}{a}|S|}\right) \leq a.$$

To prove the claim, it is enough to prove that

$$\sqrt{\frac{\beta}{a}|S|} \leq \frac{\beta}{2}|S| \Leftrightarrow \frac{2}{\sqrt{\beta a}} \leq \sqrt{|S|} \Leftrightarrow \frac{4}{\beta a} \leq |S|$$

■

In the next claim we prove that is a set is small, most likely it won't appear in a sample.

Claim 25 *Fix a dataset D, a subset $S \subseteq D$ and a scalar $0 \leq \beta \leq 1$. Assume a subset $S' \subseteq D$ of size $|S'| = \beta|D|$ is chosen uniformly at random. For any $a > 0$, if $|S| \leq \frac{a}{\beta}$ the probability that a member from S will be chosen is at most a.*

Proof The probability that a specific point be taken is β. Using union bound, the probability that at least one point from S is taken is bounded by $\beta|S| \leq a$. ■

Claim 26 *Suppose r random points are chosen uniformly at random from $C = A\dot{\cup}B$. The probability that* all *r points are in B is bounded by $\left(\frac{|B|}{|C|}\right)^r$.*

Proof Denote $n_A = |A|, n_B = |B|, n = |C| = n_A + n_B$. The random sample can be viewed as if C is ordered uniformly at random and the first r points are the sample. The probability that in a random order the first r elements are *all* from B is

$$\frac{(n-r)(n-r-1)\cdot\ldots\cdot(n-r-n_A+1)n_B!}{n!}, \tag{11}$$

the first element in A has $n-r$ locations out of n (all but the first r elements) the last element in A has $n-r-n_A+1$ locations, and the n_B members in B has no constraints in ordering them.

For any $y \geq x > a \geq 0$ it holds that $(x-a)/(y-a) \leq x/y$, thus Expression 11 is equal to

$$\frac{n_B}{n} \cdot \frac{n_B - 1}{n-1} \cdot \ldots \cdot \frac{n_B - r + 1}{n - r + 1} \leq \left(\frac{n_B}{n}\right)^r.$$

■

Corollary 27 *Fix $\delta \in (0,1)$ and $C = A\dot{\cup}B$. Suppose $\left\lceil \frac{|C|}{|A|} \log \frac{1}{\delta} \right\rceil$ random points are chosen uniformly at random from C. The probability that at least one element out of the random sample is in A is at least $1-\delta$.*

Proof Follows from Claim 26 and

$$\left(1 - \frac{|A|}{|C|}\right)^{\frac{|C|}{|A|} \log \frac{1}{\delta}} \leq \delta.$$

■

B.2. Random clustering

In this section we explore the connection between optimal clustering and one that is based on a sample of the data. Specifically, we show that clustering that is based on a random sampling is also good clustering for large optimal clusters. We focus on the case that there is a clustering with k' optimal clusters $C^* = (C_1^*, \ldots, C_{k'}^*)$. We get a random sample M of size αn and construct an optimal clustering $C^M = (C_1^M, \ldots, C_k^M)$ for those αn points with $k \geq k'$ clusters. In fact, C^M does not have to be an optimal clustering for the αn points, merely a $\Theta(1)$-approximation.

To make our claims more general, we consider an arbitrary cost that its distance function satisfy a version of a triangle inequality: there is a *constant* D such that

$$\forall u, v, w. \quad d(u, v) \leq D \cdot (d(u, w) + d(w, v)). \tag{12}$$

We call such a cost a *D-cost*.

We first prove that if an optimal clustering C_i^* is large enough, then there is a center in C^{M_1} which is a good enough center for the *entire* cluster C_i^*.

Claim 28 *Fix an a-approximation algorithm with $k \geq k'$ clusters for a random sample of size αn, D-cost, and $\delta \in (0, 1)$. For any optimal cluster C_i^*, $i \in [k']$, $|C_i^*| \geq \frac{4}{\alpha\delta}$, with probability at least $1 - \delta$ there is a center $c_{i'}^M$ in C^M such that*

$$\sum_{x \in C_i^*} d(x, c_{i'}^M) \leq \frac{5kD^2 a}{\alpha} \cdot cost(opt_{k'}).$$

Proof Denote by M the random sample of size $|M| = \alpha n$. From Claim 24, with probability at least $1 - \delta$ there are at least $\frac{\alpha |C^*|}{2}$ points in $C_i^* \cap M$. There is a cluster $C_{i'}^M \in C^M$ such that there are at least $\frac{\alpha |C^*|}{2k}$ from $C_i^* \cap C_{i'}^M$. By Inequality 3 (or Claim 34 for the k-means cost) with μ the optimal center of C_i^* it holds that

$$\begin{aligned} \sum_{x \in C_i^*} d(x, c_{i'}^M) &\leq D \sum_{x \in C_i^*} d(x, \mu) + D \sum_{x \in C_i^*} d(x, c_{i'}^M) \\ &\leq D \cdot cost(opt_{k'}) + D |C_i^*| d(\mu, c_i^*) \end{aligned}$$

Let's focus on the second term. We again use Inequality 3 and get that

$$\begin{aligned} D|C_i^*| d(\mu, c_i^*) &= \frac{2Dk}{\alpha} \cdot \frac{|C_i^*|}{2\alpha k} d(\mu, c_{i'}^M) \\ &\leq \frac{2D^2 k}{\alpha} \sum_{x \in C_i^* \cap C_{i'}^M} d(x, \mu) + \frac{2D^2 k}{\alpha} \sum_{x \in C_i^* \cap C_{i'}^M} d(x, c_{i'}^M) \\ &\leq \frac{2kD^2}{\alpha} cost(opt_{k'}) + \frac{2kD^2}{\alpha} \cdot a \cdot cost(opt_k) \\ &\leq \frac{4kD^2 a}{\alpha} cost(opt_{k'}), \end{aligned}$$

where in the second inequality we use the fact that C^M is an a-approximation and in the last inequlity we use the fact that $k \geq k'$ and thus $cost(opt_k) \leq cost(opt_{k'})$.

■

In the second auxiliary claim we prove is that for any large optimal cluster C_i^* and for any cluster in C^M with center c^M that contains $A \subseteq C_i^*$ members, its center c^M is a good center for the points in A.

Claim 29 *Fix an a-approximation algorithm with $k \geq k'$ clusters for a random sample of size αn, D-cost, and $\delta \in (0,1)$. For any optimal cluster C_i^*, $i \in [k']$, $|C_i^*| \geq \frac{4}{\alpha\delta}$, with probability at least $1-\delta$ for any cluster C_i^M with center c_i^M that contains points $A \subseteq C_i^* \cap C_i^M$ it holds that*

$$\sum_{x \in A} d(x, c_i^M) \leq \frac{5kD^2a}{\alpha} \cdot cost(opt_{k'}).$$

Proof By Claim 28 we know that with probability at least $1-\delta$ there is a center $c_{i'}^M$ in C^M such that

$$\sum_{x \in C_i^*} d(x, c_{i'}^M) \leq \frac{5kD^2a}{\alpha} \cdot cost(opt_{k'}).$$

So for any cluster $C_i^M \in C^M$ we can deduce that

$$\sum_{x \in A} d(x, c_i^M) \leq \sum_{x \in A} d(x, c_{i'}^M) \leq \sum_{x \in C_i^*} d(x, c_{i'}^M) \leq \frac{5kD^2a}{\alpha} \cdot cost(opt_{k'}).$$

where the first inequality follows from the fact that points $x \in A$ are closer to c_i^M than $c_{i'}^M$ and the second inequality holds because $A \subseteq C_i^*$. ■

In the third auxiliary claim we prove that the last claim implies that c^{M_1} is close to c_i^*.

Claim 30 *Fix an a-approximation algorithm with $k \geq k'$ clusters for a random sample of size αn, D-cost, and $\delta \in (0,1)$. For any optimal cluster C_i^*, $i \in [k']$, $|C_i^*| \geq \frac{4}{\alpha\delta}$, with probability at least $1-\delta$ for any cluster C_i^M with center c_i^M that contains points $A \subseteq C_i^* \cap C_i^M$ it holds that*

$$|A| \cdot d(c_i^M, c^*) \leq \frac{6kD^3a}{\alpha} \cdot cost(opt_{k'}).$$

Proof Use Claim 29

$$\begin{aligned} |A| \cdot d(c_i^*, c_i^M) &\leq D \sum_{x \in A} d(c_i^*, x) + d(x, c_i^M) \\ &\leq D \cdot cost(opt_{k'}) + \frac{5kD^3a}{\alpha} \cdot cost(opt_{k'}) \end{aligned}$$

■

The fourth, and the last, auxiliary claim shows that if there is a cluster in C^M that contains many points from two different optimal clusters, then these clusters can be merged, without harming the cost by much.

Claim 31 *Fix an a-approximation algorithm with $k \geq k'$ clusters for a random sample of size αn, D-cost, and $\delta \in (0,1)$. For any optimal clusters C_i^*,C_j^*, $i, j \in [k']$ with $|C_i^*|, |C_j^*| \geq \frac{8}{\alpha\delta}$, with probability at least $1-\delta$ if there is a cluster in C^M that contains at least $\zeta|C_i^*|$ points from C_i^* and at least $\eta|C_i^*|$ from C_j^*, then with probability at least $1-\delta$ it holds that*

$$cost(opt_{k'-1}) \leq \frac{13kD^5a}{\alpha \min(\zeta,\eta)} \cdot cost(opt_{k'})$$

Proof We want to bound the cost of the following clustering with $k'-1$ centers: $c_1^*, \dots, c_{k'}^*$ without c_i^* and all points in C_i^* will be assigned to c_j^*. The cost is equal to

$$\sum_{x \in C_i^*} d(x, c_j^*) + \sum_{r \neq i} \sum_{x \in C_r^*} d(x, c_r^*)$$

Let us bound the first sum using Claim 30, with probability $1-\delta$

$$\begin{aligned}
\sum_{x \in C_i^*} d(x, c_j^*) &\leq D^2 \sum_{x \in C_i^*} d(x, c_i^*) + D^2 \sum_{x \in C_i^*} d(c_i^*, c_i^M) + D \sum_{x \in C_i^*} d(c_i^M, c_j^*) \\
&= D^2 \sum_{x \in C_i^*} d(x, c_i^*) + \frac{D^2}{\zeta}\zeta|C_i^*|d(c_i^*, c_i^M) + \frac{D}{\eta}\eta|C_i^*|d(c_i^M, c_j^*) \\
&\leq D^2 cost(opt_{k'}) + \frac{6kD^5a}{\alpha\zeta} \cdot cost(opt_{k'}) + \frac{6kD^4a}{\alpha\eta} \cdot cost(opt_{k'}) \\
&\leq \frac{13kD^5a}{\alpha \min(\zeta,\eta)} \cdot cost(opt_{k'})
\end{aligned}$$

■

Appendix C. Auxiliary claims

Proof [of Lemma 10] The proof consists of simply rewriting the two expressions. The right-hand side is equal to

$$\begin{aligned}
\mathbb{E}_{j \in [n]}\left[\sum_{i=1}^{n} \|x_i - x_j\|^2\right] &= \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{n} \|x_i - x_j\|^2 \\
&= \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{n} \left(\|x_i\|^2 + \|x_j\|^2 - 2\langle x_i, x_j\rangle\right) \\
&= 2\sum_{i=1}^{n} \|x_i\|^2 - \frac{2}{n}\sum_{i,j}\langle x_i, x_j\rangle
\end{aligned}$$

The second expression is equal to twice the following expression

$$\begin{aligned}\sum_{i=1}^{n}\|x_i-\mu\|^2 &= \sum_{i=1}^{n}\left\|x_i-\frac{1}{n}\sum_{j=1}^{n}x_j\right\|^2\\ &= \sum_{i=1}^{n}\|x_i\|^2-\frac{2}{n}\sum_{i=1}^{n}\langle x_i,\sum_{j=1}^{n}x_j\rangle+\frac{1}{n}\left\|\sum_{j=1}^{n}x_j\right\|^2\\ &= \sum_{i=1}^{n}\|x_i\|^2-\frac{1}{n}\sum_{i,j}\langle x_i,x_j\rangle\end{aligned}$$

■

Claim 32 *For any scalar $q \geq 6$ and an integer $n \geq 1$ it holds that*

$$\sum_{i=1}^{n} q^i(n-i)^2 \leq 6\cdot q^{n-1}.$$

Proof

$$\begin{aligned}\sum_{i=1}^{n} q^i(n-i)^2 &= q^n\sum_{i=1}^{n}\left(\frac{1}{q}\right)^{n-i}(n-i)^2\\ &= q^n\sum_{j=0}^{n-1}\left(\frac{1}{q}\right)^{j}j^2\\ &\leq q^n\sum_{j=1}^{n-1}\left(\frac{3}{q}\right)^{j}\\ &\leq q^n\cdot\frac{3/q}{1-3/q}\\ &\leq q^n\cdot\frac{6}{q},\end{aligned}$$

where in the first equality we multiply and divide by q^n, in the second equality we reverse the order of summation, in the first inequality we use the bound

$$\left(\frac{1}{q}\right)^{j}j^2 \leq \left(\frac{3}{q}\right)^{j} \Leftrightarrow j^2 \leq 3^j$$

which is true for any $j \geq 0$, the second inequality uses the known bound for sum of a geometric series, and in the last inequality we use the bound $1 \leq 2(1-3/q)$, which is true for $q \geq 6$. ■

The next claim shows that if there are n events, each happens with probability at least $1-\delta$, then at least half of them occur together with probability at least $1-2\delta$. Specifically, for $\delta = 0.1$ we get the claim needed in the main text. For ease of notation, for any event A, we denote the indicator of A by I_A.

Claim 33 *Fix $\delta \in (0, 1)$. Suppose there are n events $A_1, \ldots, A_n$ such that for every $i \in [n]$, $\Pr(A_i) \geq 1 - \delta$. Then,*

$$\Pr\left(\sum_{i=1}^{n} I_{A_i} \geq n/2\right) \geq 1 - 2\delta.$$

Proof Let $\delta \in (0, 1)$ and events $A_1, \ldots, A_n$ with $\Pr(A_i) \geq 1 - \delta$, for every i. We want to prove that

$$\Pr\left(\sum_{i=1}^{n} I_{\neg A_i} \geq n/2\right) \leq 2\delta,$$

where $\neg A$ is the complement of A. From the assumption in the claim we know that

$$\mathbb{E}\left[\sum_{i=1}^{n} I_{\neg A_i}\right] \leq \delta n.$$

Thus, from Markov's inequality we have that

$$\Pr\left(\sum_{i=1}^{n} I_{\neg A_i} \geq \frac{1}{2\delta} \cdot \delta n\right) \leq 2\delta.$$

■

Claim 34 *For any $u, v \in \mathbb{R}^d$ it holds that*

$$\|v + u\|^2 \leq 2\|v\|^2 + 2\|u\|^2 .$$

Proof

$$\|v + u\|^2 = \|v\|^2 + \|u\|^2 + 2\langle v, u\rangle \leq \|v\|^2 + \|u\|^2 + 2\|v\|\|u\| \leq 2\|v\|^2 + 2\|u\|^2 ,$$

where the first inequality follows from Cauchy–Schwarz inequality and the second inequality follows from the inequality $0 \leq (\|u\| - \|v\|)^2 = \|v\|^2 + \|u\|^2 - 2\|v\|\|u\|$. ■

Proof of Lemma 6

Proof Take $y = \arg\min_{y \in S} \|x - y\|$ and any $y_1, y_2 \in S$ we will show that

$$\|y_1 - y_2\| \geq \|x - y\| .$$

W.l.o.g y_2 was added to S, after y_1 did. Focus at the time y_2 was added to S. Denote by $l \in S$ the closest point in S at the time to x. Then, since y_2 was added to S and not x we know that

$$\|y_2 - y_1\| \geq \|x - l\| \geq \|x - y\| .$$

■

Proof of Lemma 8

Proof

$$\begin{aligned}
\mathbb{E}_{j\in[n]}\left[\sum_{i=1}^{n} d(x_i, x_j)\right] &= \frac{1}{n}\sum_{i,j\in[n]} d(x_i, x_j) \\
&\le \frac{D}{n}\cdot\sum_{i,j\in[n]} d(x_i,\mu) + d(\mu, x_j) \\
&= 2D\cdot\sum_{i=1}^{n} d(x_i,\mu),
\end{aligned}$$

where the inequality follows from Inequality 3. ■