A Beyond-Worst-Case Analysis of Greedy k-means++[*]

Qingyun Chen
UC Santa Cruz
qchen161@ucsc.edu

Sungjin Im
UC Santa Cruz
sim9@ucsc.edu

Ryan Milstrey
UC Merced
rmilstrey@ucmerced.edu

Benjamin Moseley
Carnegie Mellon University
moseleyb@andrew.cmu.edu

Chenyang Xu
East China Normal University
cyxu@sei.ecnu.edu.cn

Ruilong Zhang
Technical University of Munich
ruilong.zhang@tum.de

Abstract

k-means++ and the related greedy k-means++ algorithm are celebrated algorithms that efficiently compute seeds for Lloyd's algorithm. Greedy k-means++ is a generalization of k-means++ where, in each iteration, a new seed is greedily chosen among multiple $\ell \geq 2$ points sampled, as opposed to a single seed being sampled in k-means++. While empirical studies consistently show the superior performance of greedy k-means++, making it a preferred method in practice, a discrepancy exists between theory and practice. No theoretical justification currently explains this improved performance. Indeed, the prevailing theory suggests that greedy k-means++ exhibits worse performance than k-means++ in worst-case scenarios. This paper presents an analysis demonstrating the outperformance of the greedy algorithm compared to k-means++ for a natural class of well-separated instances with exponentially decaying distributions, such as Gaussian, specifically when $\ell = \ln k + \Theta(1)$, a common parameter setting in practical applications.

1 Introduction

Clustering of k means is the most widely used method for data analytics. In the problem, given a set of points in a high-dimensional space along with a parameter $k > 0$ as input, we are asked to find a set of k centers in the space that minimizes the total squared distance of the points to the center set. This problem is known to be NP-hard [Aloise et al., 2009, Mahajan et al., 2012] and does not admit an approximation arbitrarily close the optimum unless P = NP [Awasthi et al., 2015, Lee et al., 2017].

While several constant-factor approximation algorithms have been found [Kanungo et al., 2002, Ahmadian et al., 2019, Grandoni et al., 2022, Cohen-Addad et al., 2022], Lloyd's heuristic [Lloyd, 1982] remains the most popular in practice due to its fast running time, simplicity, and easy adaptation to parallel and distributed settings. Lloyd's heuristic starts with k initial seeds and iteratively updates the k clusters by alternating between assigning points to their closest centroid and recomputing the centroid of each cluster.

It is well known that the quality of Lloyd's k-means clustering critically depends on the initial centroids (seeds). In fact, poorly chosen seeds can lead to significantly worse clustering results, even an unbounded cost compared to the optimum even for fixed k and n values [Arthur et al., 2007].

[*] All authors (ordered alphabetically) have equal contributions and are corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

To address this issue, [Arthur et al., 2007] proposed a very simple seeding method called k-means++. It begins by choosing the first seed uniformly at random from the data points and subsequently samples a seed with probability proportional to the squared distance of that point to the already chosen seeds. They proved that k-means++ achieves an $O(\ln k)$-approximation in expectation. k-means++ has been commonly used in conjunction with Lloyd's heuristic because the seeding is fast and it can make the solution returned by Lloyd's significantly better.

Interestingly, what is more commonly implemented is a variant of k-means++, which is called *greedy k-means++*. The greedy variant of the algorithm differs in that in each iteration, it first samples $\ell \geq 2$ candidates and chooses the candidate from them as a seed that decreases the cost function the most. This greedy algorithm is also discussed in the paper that introduced k-means++ [Arthur et al., 2007] and is implemented in popular libraries such as Scikit-learn library [Pedregosa et al., 2011] because of its superior experimental performance over k-means++.

The expectation had been that the greedy variant would yield a seeding at least as good as, if not better than, k-means++ theoretically in the worst case. Surprisingly, the greedy k-means++ was recently shown to be $\Omega(\ell \log k)$-approximate [Bhattacharya et al., 2020]. That is, strictly worse than the standard k-means++ for certain instances, which contradicts empirical findings.

The lower bound was further improved to be $\Omega(\ell^3 \log^3 k / \log^2(\ell \log k))$ [Grunau et al., 2023]. The paper additionally showed that it is $O(\ell^3 \log^3 k)$-approximate on the positive side. These results do not explain why greedy outperforms k-means++ in practice. The gap between theory and practice motivates this paper.

1.1 Our Results and Contributions

In this paper, we present the **first beyond-worst-case analysis of Greedy k-means++** for a natural class of instances to bridge the gap between theory and practice. Our analysis demonstrates that the greedy k-means++ algorithm indeed outperforms k-means++ for such instances.

Theoretical challenge. In the line of research focused on beyond-worst-case analysis of the k-means problem, a well-explored class of instances is *well-separable* inputs, where an optimal k-means clustering partitions X into k clusters and the distance between any two cluster centroids is sufficiently large compared to the variance of each cluster. Well-separable instances were widely considered in the literature [Jaiswal and Garg, 2012, Ackermann and Blömer, 2010, Braverman et al., 2011, Shechner et al., 2020]. However, even with such a simple point set, it has been theoretically proven that greedy k-means++ performs worse than k-means++ in [Bhattacharya et al., 2020].

Main theoretical result. We demonstrate that Greedy k-means++ outperforms k-means++ for a natural class of instances, which we call *regular*[2], with the following properties:

1. Points of each cluster follow an exponentially decaying distribution, such as Gaussian [Bishop, 2007, Aggarwal and Reddy, 2014]. Further, each cluster is assumed to be symmetric since k-means clustering is not well suited for highly asymmetric clusters.
2. Clusters have a similar number of points within a constant factor. This is justified by the fact that while k-means doesn't explicitly assume approximately equal-sized clusters, it is known to struggle with clusters of significantly different size [Bishop, 2013].
3. The distances between clusters are within a constant factor of k^θ where $\theta \in (0, 1/2]$, assuming that each cluster's radius (the average distance of points to the center) is $\Theta(1)$. This property is also obtained naturally: the distance between k centers sampled uniformly at random from $[0,1]^k$ is $\Theta(\sqrt{k})$ with a high probability.

In essence, the above assumptions characterize the instances where the k-means clustering method can be ideally used. For these k-means clustering-friendly instances, we show that Greedy k-means++ with $\ell = \ln k + \Theta(1)$ is $O((\ln \ln k)^2)$-approximate for the above regular instances, as opposed to k-means++ that is shown to be $\Omega(\ln k)$-approximate. Provided that $\ell = \Theta(\ln k)$ is a common choice in practice—for example, the Scikit-learn library [Pedregosa et al., 2011] sets $\ell = \lceil \ln k \rceil + 2$, our result implies that the commonly used Greedy has a better theoretical guarantee over k-means++, even asymptotically.

[2] Later, we will more formally define the instances to consider and call them EWW.

Analysis overview. The high-level analysis idea is as follows. Let us say that a cluster is covered if a point within the cluster has been chosen as a seed. Since clusters are sufficiently far from each other and have similar sizes, it is advantageous to cover all clusters.

For simplicity, suppose that we are at the beginning of the κ-th iteration and have successfully covered the $\kappa - 1$ clusters without wasting any previous seeds. Further, assume that the chosen seeds are close to the cluster centers, which is ensured by the fact that clusters have exponentially decaying tails in distance, and therefore most sampled points are not too far from their centers. Here, it is used that if a point is conditioned on being sampled from a specific cluster, the sample more or less follows a uniform sampling from the cluster since the clusters are sufficiently far from one another.

Then, we can show that given multiple sampled candidates, the greedy algorithm chooses one from an uncovered cluster to decrease the k-means objective the most. Thus, as long as the greedy algorithm finds at least one candidate point from an uncovered cluster out of $\ell = \Theta(\ln k)$ candidates, it successfully covers an additional cluster in the current κ-th iteration.

The greedy algorithm may select a candidate point further from the centers when given multiple such candidate points, since it may prefer a point that is close to many clusters simultaneously. However, since the clusters have exponentially decaying tails, we can show that even the worst candidate point has a distance of at most $O(\ln \ln k)$ from its respective center in expectation. In contrast, we show that for regular instances, k-means++ is highly likely to fail to cover many clusters because it only samples one point in each iteration.

When clusters are very far apart from each other. Finally, we briefly discuss when the regular instance has large distances between the cluster centers, more formally when $\theta \geq 1/2$. In this case, it becomes hard to show that the greedy algorithm performs better in terms of the k-means objective. This is not surprising. Intuitively, when clusters are very far away from each other, each cluster can essentially be seen as a single point, and therefore there is very little room for k-means++ to make mistakes. However, we can still show that the greedy algorithm has a higher chance of covering all clusters than k-means++. Intuitively, the initial seeds having covered all clusters will likely result in a good clustering by a subsequent run of Lloyd's algorithm. Thus, even in this case, we indirectly show the advantage of the greedy algorithm over k-means++.

Experiments. Since it is well known that the greedy algorithm outperforms k-means++ in practice and empirical studies, we rather focus on our experiments on tracking how the algorithm makes choices over iterations towards a better seeding. We create synthetic data sets using various distributions and study how the algorithm makes progress in terms of covering new clusters over iterations, not only tracking how the objective changes. The experiments show that the greedy algorithm outperforms k-means++ in both decreasing the objective and covering new clusters. Thus, the experiments further corroborate the greedy algorithm's better performance, together with the theoretical analysis.

1.2 Other Related Work

While k-means clustering is NP-hard to solve optimally, for any $\epsilon > 0$, the problem admits a $(1 + \epsilon)$-approximation when either k or d (the number of dimensions) is a constant. Feldman et al. [2007], Kumar et al. [2004]. Recent works show that k-means++ can also be used with a small number of steps of a local search algorithm Lattanzi and Sohler [2019], Choo et al. [2020] to yield $O(1)$-approximations. This result is further improved to $(9 + \epsilon)$ by Beretta et al. [2023], which matches the best approximation for the local search algorithm Kanungo et al. [2002]. Sketching can be used to compress the input data into a compact subset of points, called a coreset. This allows for faster clustering by running the algorithm on the coreset instead of the original data (e.g., Har-Peled and Mazumdar [2004], Chen [2009]).

There is currently no theoretical analysis of Greedy k-means++ beyond the works by Bhattacharya et al. [2020], Grunau et al. [2023]. For a comparative study of seeding methods, see Celebi et al. [2013]. They recommend a value of ℓ (number of candidates sampled per iteration) proportional to the logarithm of k (number of clusters). The Scikit-learn library specifically sets $\ell = \lceil \ln k + 2 \rceil$ Pedregosa et al. [2011].

Except for the classical Greedy k-means++ and k-means++, several variants have also been studied. Aggarwal et al. [2009] show that k-means++ is $O(1)$-approximation with constant probability if it allows selecting $O(k)$ centers. This bicriteria approximation is further improved by Makarychev et al.

[2020], Wei [2016]. Balcan et al. [2018] suggest seeding the initial centers via D^α-sampling, which generalizes the k-means++ algorithm ($\alpha = 2$). Bamas et al. [2024] analyze the new seeding method and show that D^α-sampling admits better approximation than D^2-sampling under specific instances.

The k-means clustering has also been studied in various settings, including distributed Bahmani et al. [2012], streaming Ailon et al. [2009], and dynamic environments Bhattacharya et al. [2024]. In particular, Bahmani et al. [2012] extends k-means++ to a distributed setting.

1.3 Organization

In the following section, we recall the greedy k-means++ and k-means++ algorithms and formally define the instances we will consider throughout this paper. To make our presentation transparent, we will only show our results for more restricted instances requiring fewer parameters, deferring the analysis of the more general instances to the appendix. Then, we analyze the greedy and k-means++ when the distances between clusters are not too large in Section 3, 4. The other case is handled in Section 5. After presenting experiments in Section 6, we conclude the paper.

2 Preliminaries

This section formally defines the k-means clustering problem, along with notations and background.

k-Means Clustering. Consider an m-dimensional Euclidean space $\mathbb{R}^m$. For a point $x \in \mathbb{R}^m$, its connection cost to a point set $C \subseteq \mathbb{R}^m$ is defined as the squared distance of x to its closest point in C, i.e., $\varphi(x, C) := \min_{c \in C} ||x - c||_2^2$. In the k-means problem, we are given a set of n points $X \subseteq \mathbb{R}^m$ as well as a parameter $k \in \mathbb{N}_{>0}$, and the goal is to find a set of k centers $S \subseteq \mathbb{R}^d$ that minimizes the total connection cost of points in X to S, i.e., $\varphi(X, S) := \sum_{x \in X} \varphi(x, S)$.

For a given point set X, let $\{C_i\}_{i \in [k]}$ denote the k clusters in the optimal solution, and let $\{\mu_i\}_{i \in [k]}$ represent the corresponding cluster centers. We define $C_i(r)$ as the set of points in C_i that are at a distance r from μ_i.

We first formally present the statements of greedy k-means++ in Algorithm 1. The k-means++ algorithm is a special case of Algorithm 1 when $\ell = 1$. Both of the two algorithms run iteratively. In each iteration, k-means++ samples one candidate from the probability distribution $\{\varphi(x, S)/\varphi(X, S)\}$ while greedy k-means++ samples $\ell > 1$ candidates and pick the one with the minimum connection cost. Since Greedy k-means++ commonly uses $\ell = \ln k + \Theta(1)$ Pedregosa et al. [2011], we will assume $\ell = \ln k$ and k is sufficiently large unless stated otherwise.

Algorithm 1 Greedy k-means++ Initialization Arthur et al. [2007]

```
Input: A point set $X \subseteq \mathbb{R}^m$ and parameters $k > 0, \ell = \ln k$[3].
Output: A center set $S$ that serves as the initial centers of Lloyd's heuristic.
Independently and uniformly sample $\ell$ points $x_1, \ldots, x_\ell \in X$.
Greedily pick $x := \arg\min_{x_i \in \{x_1, \ldots, x_\ell\}} \varphi(X, \{x_i\})$ and set $S \leftarrow \{x\}$.
for $t = 2, \ldots, k$ do
    Sample $\ell$ points $x_1, \ldots, x_\ell \in X$ independently (with replacement) with probability $\frac{\varphi(x,S)}{\varphi(X,S)}$.
    Greedily pick $x := \arg\min_{x_i \in \{x_1, \ldots, x_\ell\}} \varphi(X, S \cup \{x_i\})$, breaking ties arbitrarily.
    Set $S \leftarrow S \cup \{x\}$.
end for
return $S$.
```

The paper analyzes these two algorithms on the input point set X satisfying the three properties:

- *Exponentially Distributed:* For each cluster C_i, the density of points decreases exponentially as the distance from the center increases. Specifically, $\frac{|C_i(r)|}{|C_i|} = \frac{1}{b} \cdot e^{-r/b}$, for a constant $b > 0$.
- *Well Separable*: The minimum distance d between any two centers of the optimal clusters be sufficiently large relative to the cluster distribution parameter b, i.e., $d = k^\theta \cdot b$, for a constant $\theta > 0$.

[3] We assume that $\ln k > 1$ is an integer for notational convenience, instead of using $\lceil \ln k \rceil$ explicitly.

- *Well Spread*: The optimal clusters are roughly homogeneous, with the number of points in each cluster and the distances between clusters differing by at most a constant factor.

Define a point set that satisfies the above properties as an EWW point set. We remark that our analysis applies to any subexponential-tailed distribution, such as Gaussian and sub-Gaussian distributions, with a mild assumption. The details are deferred to Appendix C. For simplicity and readability, we assume throughout the analysis that each cluster has the same size and equal pairwise distances. We remark that the analysis can be easily extended to the case where these quantities differ by at most a constant factor. These extensions can also be found in Appendix C.

We now present several useful observations concerning the structure of EWW point sets, which can be readily derived through straightforward mathematical calculations.

Observation 1. *The optimal total connection cost is achieved by selecting each cluster center μ_i, resulting in an objective:* $\mathsf{OPT} = n\int_0^\infty \frac{r^2}{b} \cdot e^{-r/b} dr = 2b^2 n$, *where n is the number of points.*

Observation 2. *Consider a cluster C_i with its center μ_i. For a point x located at distance h from μ_i, the total connection cost of all points in C_i to x is $(2b^2 + h^2) \cdot |C_i|$.*

3 Approximation Guarantee of Greedy k-Means++

In this section, we analyze the performance of the greedy k-means++ algorithm on EWW point sets and aim to show the following:

Theorem 1. *Given any EWW point set X, the greedy k-means++ algorithm admits an expected approximation ratio of $O((\ln\ln k)^2)$.*

Proof Outlines. Due to the exponentially distributed property, we can show that, with very high probability, the points sampled by the algorithm are located close to the optimal centers $\{\mu_i\}_{i\in[k]}$. In the following, we define a *concentration ball* for each cluster and prove that we may assume the algorithm never samples points outside these balls. In particular, the chance the algorithm samples a point outside these balls is very low and, due to this, the total contribution to the expected objective is small in this case. Under this concentration assumption, we classify the clusters $\{C_i\}_{i\in[k]}$ into two types at each iteration: *covered* clusters (those for which the algorithm has already selected a point within the corresponding concentration ball), and *uncovered* clusters (the rest). Due to the greedy nature of the algorithm, it always prefers candidate points from uncovered clusters over those from covered clusters. Leveraging the *well-separability* and *well-spreadness*, we show that with high probability, the algorithm covers a new cluster in most iterations, thereby achieving the desired approximation ratio.

We remark that, while the above captures the high-level strategy of our analysis, applying only these techniques can only give an approximation ratio of $O((\ln k)^2)$. To improve the ratio to $O((\ln\ln k)^2)$, we further exploit the independence among clusters induced by well separability and conduct a more careful analysis. We begin by introducing the definition of a concentration ball and the corresponding lemma.

Definition 1 (Concentration Ball)**.** *We define the* concentration radius $\delta := 4b(1+\theta)\ln k$, *where θ and b are input parameters. For each cluster C_i with center μ_i, we define its* concentration ball $\sigma(C_i)$ *as the set of all points in C_i whose distance to μ_i is at most δ.*

Lemma 1 (Concentration Lemma)**.** *Let $\mathcal{A}$ denote the event that greedy k-means++ samples at least one candidate point outside the concentration balls during any iteration. Given any EWW point set, the probability that $\mathcal{A}$ occurs is at most $1/k$. Furthermore, the contribution of this event to the expected objective can be bounded:* $\Pr[\mathcal{A}] \cdot \mathbb{E}[\mathsf{OBJ} \mid \mathcal{A}] \le \frac{n}{k}$, *where $\mathbb{E}[\mathsf{OBJ} \mid \mathcal{A}]$ denotes the expected objective value (i.e., total connection cost) conditioned on the occurrence of $\mathcal{A}$. Furthermore, this upper bound holds as well when $\mathcal{A}$ is k-means++.*

As $\mathsf{OPT} = \Theta(n)$ (see Observation 1), the above lemma implies that the total contribution of points outside the concentration balls is modest compared to the optimal cost. Hence, throughout the remainder of this paper, we adopt the *concentration assumption*, under which no point outside the concentration balls is sampled by the algorithm, whether it is greedy k-means++ or k-means++.

We say that a feasible solution *covers* a cluster C_i if it includes at least one point from its concentration ball $\sigma(C_i)$. We claim the following:

Lemma 2. *Under the concentration assumption, for each cluster C_i , we have:*

(2a) If greedy k-means++ does not cover this cluster, then its total connection cost is $\Omega(k^{2\theta} \cdot |C_i|)$.

(2b) If greedy k-means++ covers this cluster, then the total connection cost for C_i does not exceed $O((\ln k)^2 \cdot |C_i|)$, and further, its expectation is $O((\ln \ln k)^2 \cdot |C_i|)$ and $\Omega(|C_i|)$.

As established in Lemma 2, under the well-spread assumption, achieving the approximation guarantee stated in Theorem 1 requires that the number of uncovered clusters is at most $O(k^{1-2\theta} \cdot (\ln \ln k)^2)$. Otherwise, the total connection cost contributed by the uncovered clusters would exceed $(\ln \ln k)^2 \cdot \mathsf{OPT}$, violating the desired bound. The next lemma establishes the probability of covering a new cluster, which will later be used to determine the number of uncovered clusters.

Lemma 3. *In each iteration $t \leq k - O(k^{1-2\theta} \cdot (\ln \ln k)^2)$, the greedy k-means++ algorithm covers a new cluster with probability at least $1 - 1/k^{2\theta+2}$.*

Proof. We first leverage (2a) of Lemma 2 to show that, with high probability, the greedy k-means++ algorithm covers a new cluster in each iteration up to iteration $k - O(k^{1-2\theta} \cdot (\ln k)^2)$. Then, by applying (2b) of Lemma 2 along with the Chernoff bound, we demonstrate that the algorithm continues to cover a new cluster in each iteration, with high probability, even during the range of iterations from $k - O(k^{1-2\theta} \cdot (\ln k)^2)$ to $k - O(k^{1-2\theta} \cdot (\ln \ln k)^2)$. Intuitively, the reason we cannot directly apply the Chernoff bound is that, when doing so, we require the connection cost upper bound of each individual covered cluster to be significantly smaller than the expected total connection cost of the covered clusters, which only holds when the greedy is at at a later iteration, say for $t \geq k/2$.

First, consider an iteration $k \leq t_0 := k - 2^{2\theta+5} \cdot k^{1-2\theta} \cdot (\ln k)^2$. Lemma 2 implies that the greedy comparison step always prefers points from uncovered concentration balls over those from already covered ones. This is because if the greedy selects a point from an uncovered concentration ball (by (2a)), it decreases the objective by $\Omega(n/k)k^{2\theta}$ while if it selects from a covered concentration ball, it only decreases the objective by $O(n/k)(\ln k)^2$ (by (2b)). Thus, it follows that, in iteration t, if at least one of the $\ln k$ sampled candidates is from an uncovered concentration ball, then the algorithm is guaranteed to cover a new cluster in that iteration. Suppose that p clusters have already been covered by S_{t-1} at the beginning of iteration t (so $p < t$). By (2b) of Lemma 2 and the well-spread property, the total connection cost contributed by the covered clusters is at most $\frac{n}{k} \cdot p \cdot (\ln k)^2$, while the total connection cost from the uncovered clusters is at least $\frac{n}{k} \cdot (k-p) \cdot k^{2\theta}$, where we omit constant factors for simplicity—such constants do not affect our analysis. Therefore, the probability[4] that the algorithm fails to sample any point from the uncovered concentration balls is at most

$$\left(\frac{p \cdot (\ln k)^2}{p \cdot (\ln k)^2 + (k-p) \cdot k^{2\theta}} \right)^{\ln k} \leq 1/k^{2\theta+5} \text{ when } p \leq t_0.$$

Next, we analyze the time period between iteration t_0 and $t_1 := k - 2^{2\theta+5} \cdot k^{1-2\theta} \cdot (\ln \ln k)^2$. From the earlier analysis and by applying a union bound, we know that with probability at least $1 - 1/k^{2\theta+4}$, the algorithm has already covered t_0 clusters within the first t_0 iterations. Then, according to Lemma 2, conditioned on this event, the expected connection cost of these covered clusters is lower bounded by $\Omega(\frac{n}{k} \cdot t_0)$, which is asymptotically much larger than the upper bound on the connection cost for any single cluster, $O\left(\frac{n}{k} \cdot (\ln k)^2\right)$. This suggests the connection cost of the covered clusters is well-concentrated.

We shall upper bound the total connection cost of the covered clusters in the first t_0 iterations via the concentration bound. Note that each cluster can be approximately treated as being sampled uniformly based on the analysis in the proof of Lemma 2, which enables us to use concentration inequalities.

Applying the concentration bound, we get the following claim; the proof is deferred to the appendix.

Claim 1. *With probability at least $1 - \exp\left(\Theta(1) \cdot (-k) \cdot \left(\frac{\ln \ln k}{\ln k}\right)^2\right)$, the total connection cost of the clusters covered in the first t_0 iterations is at most $2b^2 \cdot \frac{n}{k} \cdot t_0 \cdot (\ln \ln k)^2$.*

[4] As mentioned earlier, for simplicity, we assume that each cluster is of equal size and that they are equidistant from one another. We note that if the clusters are not exactly equal in size but differ by only a constant factor, the probability will increase by at most a constant, and the overall order of magnitude will remain the same.

Consider any iteration $t \in (t_0, t_1]$. The algorithm can cover at most $t_1 - t_0$ new clusters during the interval from iteration t_0 to t_1, and each newly covered cluster contributes at most $O\left(\frac{n}{k} \cdot (\ln k)^2\right)$ to the connection cost. Thus, with a probability of at least

$$\left(1 - \frac{1}{k^{2\theta+4}}\right)\left(1 - \exp\left(\Theta(1) \cdot (-k) \cdot \left(\frac{\ln\ln k}{\ln k}\right)^2\right)\right) ,$$

the total connection cost of the already covered clusters at the beginning of iteration t is at most

$$A := 2 \cdot \frac{n}{k} \cdot \left(b^2 \cdot t_0 \cdot (\ln\ln k)^2 + (t_1 - t_0) \cdot (\ln k)^2\right) .$$

Note that the total connection cost from the uncovered clusters is at least $B := \frac{n}{k} \cdot (k - t_0) \cdot k^{2\theta}$. Similar to the analysis in the previous case, the probability that the algorithm fails to sample any point from the uncovered concentration balls is at most $(A/(A+B))^{\ln k}$. This implies that the probability that greedy k-means++ fails to cover a new cluster is at most $1/k^{2\theta+2}$. □

Proof of Theorem 1. By Lemma 3 and a union bound, we have that, with probability at least $1 - 1/k^{2\theta+1}$, the greedy k-means++ algorithm covers $k - O\left(k^{1-2\theta} \cdot (\ln\ln k)^2\right)$ clusters and achieves an approximation ratio of $O((\ln\ln k)^2)$. If this case does not occur, we can simply use an upper bound on the objective of $O(n \cdot k^{2\theta})$. Taking expectation over both cases, we obtain an expected approximation ratio of $O((\ln\ln k)^2)$. □

4 Approximation Lower Bound of k-Means++

This section analyzes the k-means++ algorithm and establishes a lower bound $\Omega(\ln k)$ on its approximation ratio. This lower bound highlights a gap between the performance of the greedy k-means++ and standard k-means++ algorithms: while the latter suffers from an $\Omega(\ln k)$ lower bound, the former achieves an $O((\ln\ln k)^2)$ approximation ratio on the same instances, demonstrating the theoretical advantage.

Theorem 2. *Given any EWW point set X with parameter $\theta \in (0, 1/2]$, the k-means++ algorithm has an expected approximation ratio of $\Omega(\ln k)$.*

Our proof strategy mirrors that used for greedy k-means++, where we analyze the expected number of uncovered clusters to derive bounds on the approximation ratio. Specifically, to establish a lower bound for the algorithm, we aim to show that with non-negligible probability, k-means++ selects points from already covered concentration balls in certain iterations (recall that k-means++ samples only one point per iteration, whereas greedy k-means++ samples $\ln k$ candidates). To this end, we require a lemma symmetric to Lemma 2, which provides a lower bound on the probability that the algorithm samples from an already covered concentration ball.

Lemma 4. *Under the concentration assumption, for each cluster C_i , we have:*

(4a) *If k-means++ does not cover this cluster, then its total connection cost is $\Theta(k^{2\theta} \cdot |C_i|)$.*

(4b) *If k-means++ covers this cluster using exactly one center—that is, the final solution includes exactly one point from $\sigma(C_i)$—then the total connection cost for C_i is $\Omega(|C_i|)$.*

Proof of Lemma 4: This proof follows a similar analysis to that in Lemma 2. By Observation 2, the total connection cost of a cluster C_i is $(2b^2 + h^2) \cdot |C_i|$, where h denotes the distance from the cluster center μ_i to the current set of selected centers S. When the cluster is not yet covered, $h \in [d - \delta,\ d + 2\delta]$, which is of order $\Theta(k^\theta)$; whereas once the cluster is covered, h can be as small as 0. This completes the proof of the lemma. ■

Proof of Theorem 2. Lemma 4 shows that if the number of uncovered clusters is p, then the objective value is at least $\frac{n}{k} \cdot p \cdot k^{2\theta}$, omitting constant factors for simplicity. Therefore, to establish a lower bound of $\Omega(\ln k)$ on the approximation ratio, it suffices to show that, in expectation, k-means++ leaves $\Omega(\ln k \cdot k^{1-2\theta})$ clusters uncovered. This also explains why we require $\theta \in (0, 1/2]$: otherwise, $\ln k \cdot k^{1-2\theta}$ would be subconstant, rendering the argument meaningless.

We partition all possible outcomes of the algorithm into two cases based on the number of uncovered clusters: (1) the final number of uncovered clusters is at least Δ, and (2) the final number of uncovered clusters is less than Δ, where $\Delta = \ln k \cdot k^{1-2\theta}$. Clearly, in all outcomes falling into the first case, the approximation ratio is $\Omega(\ln k)$. Next, we analyze the second case and show that, in expectation, the number of uncovered clusters remains $\Omega(\Delta)$.

Consider an arbitrary iteration t. In the second case, where the final number of uncovered clusters is less than Δ, the number of clusters already covered by the solution S_{t-1} at the beginning of this iteration must be at least $t - \Delta$. Otherwise, even if every subsequent iteration covers a new cluster, the final number of uncovered clusters would exceed Δ, contradicting the assumption. Then, by the pigeonhole principle, at least $t - 2\Delta$ clusters must be covered by exactly one center—that is, S_{t-1} contains exactly one point from each of these clusters. By Lemma 4 and the well-spread property, the total connection cost of the covered clusters is at least $\frac{n}{k} \cdot (t - 2\Delta)$, while the total connection cost of the uncovered clusters is at most $\frac{n}{k} \cdot (k - t + \Delta) \cdot k^{2\theta}$. Therefore, in each iteration $t > 2\Delta$, the probability that the k-means++ algorithm fails to cover a new cluster is at least $\frac{t-2\Delta}{(t-2\Delta)+(k-t+\Delta)\cdot k^{2\theta}}$.

We compute the expected number of uncovered clusters by summing the failure probabilities across all iterations (conditioned on the second case). Specifically, we have:

$$\mathbb{E}[\text{number of uncovered clusters}] \geq \sum_{t>2\Delta}^{k} \frac{t-2\Delta}{(t-2\Delta)+(k-t+\Delta)\cdot k^{2\theta}}$$

$$\geq \sum_{t\geq k/2}^{k} \frac{t-2\Delta}{(t-2\Delta)+(k-t+\Delta)\cdot k^{2\theta}} \geq \sum_{t\geq k/2}^{k} \frac{k/4}{k/4+(k-t+\Delta)\cdot k^{2\theta}} \qquad (\Delta = o(k))$$

$$= k^{1-2\theta} \cdot \sum_{t\geq k/2}^{k} \frac{1}{k^{-2\theta}+4(k-t+\Delta)} \geq k^{1-2\theta} \ln\left(\frac{k/2+\Delta}{\Delta}\right)$$

$$\geq k^{1-2\theta} \ln\left(\frac{k^{2\theta}}{2\ln k}\right) = \Omega(k^{1-2\theta} \ln k) \ ,$$

which implies that the expected number of uncovered clusters is $\Omega(\Delta)$ and completes the proof. □

5 Analysis of Covering Probability

Theorem 1 and Theorem 2 demonstrate that, on the EWW point set with parameter $\theta \in (0, 1/2]$, the greedy k-means++ algorithm achieves a better approximation ratio than the standard k-means++ algorithm. The intuition is that when $\theta \in (0, 1/2]$, the optimal clusters are not yet well-separated, so the probability that k-means++ fails to cover a new cluster in a given iteration remains relatively high. In contrast, greedy k-means++ can exponentially reduce this failure probability through multiple samples per iteration.

As θ increases further and the optimal clusters become more widely separated, the failure probability for standard k-means++ correspondingly decreases, reducing the approximation gap between the two algorithms. This section formally addresses such cases, showing that even in these settings, greedy k-means++ remains theoretically superior to standard k-means++ from a certain perspective.

Theorem 3. *Given any EWW point set with parameter $\theta > 1/2$, the probability that greedy k-means++ covers all optimal clusters is greater than that of k-means++.*

One might find this theorem intuitively trivial. Since greedy k-means++ performs multiple samples per iteration, it should naturally have a higher probability of covering a new cluster than k-means++, which would suggest the theorem's correctness. However, this reasoning strictly holds only when both algorithms share the same set of selected centers S, which we cannot guarantee. In fact, during the execution of greedy k-means++ and k-means++, the distribution over all possible center sets S_t at each iteration t may differ significantly, which makes the proof of the theorem non-trivial.

Observe that if an algorithm fails to cover all optimal clusters, it must have selected at least two points from the same optimal cluster. Therefore, the probability that an algorithm covers all optimal clusters is equal to 1 minus the probability that there exists at least one iteration in which the algorithm selects a point from an already covered cluster.

To prove Theorem 3, we analyze the probabilities that greedy k-means++ and k-means++ encounter such a bad event. Specifically, we establish a lower bound on that probability for k-means++ (Lemma 5) and an upper bound for greedy k-means++ (Lemma 6), and finally show that the former is greater than the latter. Let event $\mathcal{B}$ denote the bad event that the algorithm selects a point from an already covered cluster. We claim the following.

Lemma 5. *The probability that k-means++ encounters $\mathcal{B}$ is at least $\frac{k-1}{k-1+k^{2\theta}}$.*

Proof of Lemma 5: We partition the bad event into two sub-events based on the time at which $\mathcal{B}$ first occurs: (1) k-means++ encounters $\mathcal{B}$ before the last iteration k, and (2) k-means++ first encounters $\mathcal{B}$ at the last iteration k. We denote these two sub-events as $\mathcal{P}$ and $\mathcal{Q}$, respectively. By expanding the conditional probability of the second sub-event, we derive a lower bound on the probability that k-means++ encounters $\mathcal{B}$:

$$\Pr[k\text{-means++ encounters } \mathcal{B}] = \Pr[\mathcal{P}] + \Pr[\mathcal{Q}] = \Pr[\mathcal{P}] + \Pr[\neg\mathcal{P}] \cdot \Pr[\mathcal{Q} \mid \neg\mathcal{P}] \geq \Pr[\mathcal{Q} \mid \neg\mathcal{P}].$$

Conditioned on $\neg\mathcal{P}$, the notion of "first" in event $\mathcal{Q}$ is not essential—$\Pr[\mathcal{Q} \mid \neg\mathcal{P}]$ simply equals the probability that k-means++ samples a point from one of the $k-1$ already covered clusters. By Lemma 4, the total connection cost of the already covered clusters is at least $\frac{n}{k} \cdot (k-1)$, while the total connection cost of the last uncovered cluster is at most $\frac{n}{k} \cdot k^{2\theta}$, omitting constant factors for simplicity. Thus, the probability that k-means++ encounters event $\mathcal{B}$ can be lower bounded by $\frac{k-1}{k-1+k^{2\theta}}$. ■

Lemma 6. *The probability that greedy k-means++ encounters $\mathcal{B}$ is at most $\left(\frac{e\cdot(k-1)\cdot(\ln k)^2}{(k-1)\cdot(\ln k)^2+k^{2\theta}}\right)^{\ln k}$.*

Proof of Lemma 6: To upper bound the probability for greedy k-means++, we partition the bad event into k sub-events $\{\mathcal{P}_t\}_{t\in[k]}$, where $\mathcal{P}_t$ denotes the event that greedy k-means++ encounters $\mathcal{B}$ for the first time in iteration t. Similarly, we then expand the probability of $\mathcal{P}_t$ using conditional probability:

$$\Pr[\text{greedy } k\text{-means++ encounters } \mathcal{B}] = \sum_{t\in[k]} \Pr[\mathcal{P}_t]$$

$$= \sum_{t\in[k]} \Pr[\neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{t-1})] \cdot \Pr[\mathcal{P}_t \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{t-1})] \leq k \cdot \Pr[\mathcal{P}_k \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{k-1})]$$

where the last inequality uses the fact that $\Pr[\mathcal{P}_t \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{t-1})]$ attains its maximum at $t = k$.

The term $\Pr[\mathcal{P}_k \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{k-1})]$ simply equals the probability that greedy k-means++ samples all $\ln k$ candidates from one of the $k-1$ already uncovered clusters. By Lemma 2, the total connection cost of the already covered cluster is at most $\frac{n}{k} \cdot (k-1) \cdot (\ln k)^2$, while the total connection cost of the last uncovered cluster is at most $\frac{n}{k} \cdot k^{2\theta}$, omitting constant factors for simplicity. Thus, the probability that greedy k-means++ encounters $\mathcal{B}$ can be upper bounded by

$$k \cdot \left(\frac{(k-1)\cdot(\ln k)^2}{(k-1)\cdot(\ln k)^2 + k^{2\theta}}\right)^{\ln k} = \left(\frac{2\cdot(k-1)\cdot(\ln k)^2}{(k-1)\cdot(\ln k)^2 + k^{2\theta}}\right)^{\ln k} .$$

■

Proof of Theorem 3. Lemma 5 and Lemma 6, through a series of mathematical calculations, directly establish the theorem. More specifically, when $\theta > 1/2$, the lower bound on the failure probability for k-means++ is $\Theta(k^{1-2\theta})$, while the upper bound for greedy k-means++ is $\Theta\left(k^{e\ln\ln k+(1-2\theta)\ln k}\right)$. As the former asymptotically dominates the latter, we can conclude that greedy k-means++ has a higher probability of covering all optimal clusters than k-means++. □

6 Experiments

Our experiments are conducted on a machine with Processor 11th Gen Intel(R) Core(TM) i5-1135G7 2.40GHz, 1382 Mhz, 4 Core(s), 8 Logical Processor(s) and 12 GB RAM. We evaluate the performance of greedy k-means++ and k-means++ on 3 datasets.

Our goal is to demonstrate that the theory is predictive of practice. Our experiments correspond to our theoretical model and remark that there is extensive empirical work on both k-means++ and the greedy variant previously known.

Input Data. We conduct experiments on synthetic datasets. To generate a dataset, we first fix a distribution. In the experiment, we use the exponential, half-normal (the absolute variant of the Gaussian), and Lomax (heavy-tail sub-exponential) distributions for different datasets. We first sample k centers in $\mathbb{R}^k$ uniformly at random from a unit hypercube. Then we sample the radius for each cluster uniformly at random from $(0, 2)$. The number of points of each cluster is uniformly sampled from $[64, 256]$. To generate the points of a cluster, we choose uniform random points whose distance to the center follows the fixed distribution. Since the centers are sampled from a unit hypercube, it is well-separable because the distances between every two centers are $\sqrt{k}$. The generation of the points guarantees that it is well-spread.

Experiments. We consider two different metrics – the k-means objective and coverage probability. For greedy k-means++, we sample $\lceil \ln k \rceil + 2$ candidates in every iteration. The experiment involves 100 repetitions. We compare the average objective and coverage probability in every iteration. The average objective in an iteration is the average k-means objective using the currently chosen centers in 100 repetitions. Similarly, the average coverage probability is the probability that a new center is covered in 100 repetitions. We show the results over iterations for $k = 16$ in Figure 1 (refer to Appendix D for different k). We can see the superior performance of greedy k-means++ over k-means++ under two different measures, which validates our theory that greedy k-means++ outperforms k-means++.

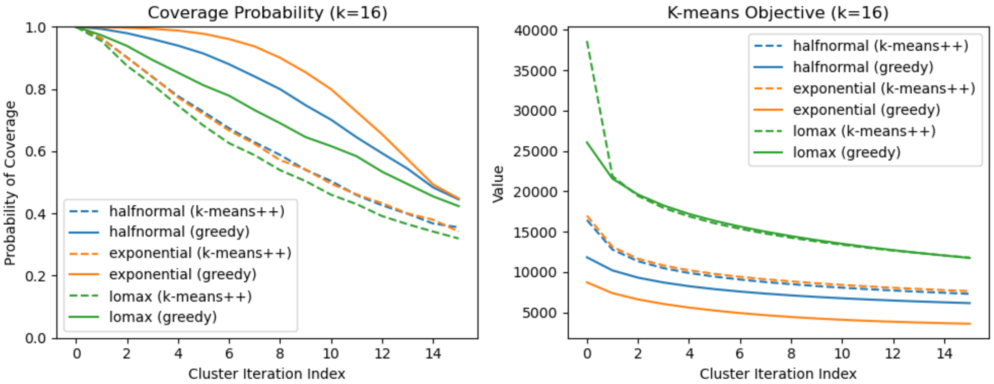

Figure 1: Coverage probability and k-means objective over iterations for $k = 16$.

7 Conclusions

In this paper, we presented the first beyond-worst-case analysis of the greedy k-means++ algorithm. We conclude the paper with some open problems. Our analysis assumes that the greedy algorithm samples $\ln k + \Theta(1)$ candidate points per iteration. While this is commonly used in practice, sampling a constant number of candidates could still place the greedy ahead of k-means++. Our current analysis falls short of showing this and studying the greedy's performance when $\ell = o(\ln k)$ could be interesting. Also, it could be plausible that one can prove the greedy algorithm has a better approximation ratio than $O((\ln \ln k)^2)$ for the EWW instances. Finally, it would be very interesting to discover new algorithms that improve the greedy algorithm now that we have a theoretical understanding of it in a beyond-worst-case setting.

Acknowledgements

Chenyang Xu was supported by the National Natural Science Foundation of China (No. 62302166) and the National Key Research Project of China under Grant No. 2023YFA1009402. Qingyun Chen, Sungjin Im, and Ryan Milstrey were supported in part by NSF grants CCF-2535599, CCF-2537126, and CCF-2423106. Benjamin Moseley was supported in part by a Google Research Award and NSF grants CCF-2121744 and CCF-1845146.

References

Marcel R Ackermann and Johannes Blömer. Bregman clustering for separable instances. In *Scandinavian Workshop on Algorithm Theory*, pages 212–223. Springer, 2010.

Ankit Aggarwal, Amit Deshpande, and Ravi Kannan. Adaptive sampling for k-means clustering. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 15–28. Springer, 2009.

Charu C. Aggarwal and Chandan K. Reddy, editors. *Data Clustering: Algorithms and Applications*. CRC Press, 2014. ISBN 978-1-46-655821-2. URL http://www.crcpress.com/product/isbn/9781466558212.

Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, 49(4): FOCS17–97, 2019.

Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/4f16c818875d9fcb6867c7bdc89be7eb-Paper.pdf.

Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75:245–248, 2009.

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pages 1027–1035, 2007.

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

Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proc. VLDB Endow.*, 5(7):622–633, 2012. doi: 10.14778/2180912.2180915. URL http://vldb.org/pvldb/vol5/p622_bahmanbahmani_vldb2012.pdf.

Maria-Florina F Balcan, Travis Dick, and Colin White. Data-driven clustering via parameterized lloyd's families. *Advances in neural information processing systems*, 31, 2018.

Etienne Bamas, Sai Ganesh Nagarajan, and Ola Svensson. Analyzing d^{α} seeding for k-means. In *Forty-first International Conference on Machine Learning*, 2024.

Lorenzo Beretta, Vincent Cohen-Addad, Silvio Lattanzi, and Nikos Parotsidis. Multi-swap k-means++. *Advances in Neural Information Processing Systems*, 36:26069–26091, 2023.

Anup Bhattacharya, Jan Eube, Heiko Röglin, and Melanie Schmidt. Noisy, Greedy and Not so Greedy k-Means++. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:21, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-162-7. doi: 10.4230/LIPIcs.ESA.2020.18. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2020.18.

Sayan Bhattacharya, Martín Costa, Silvio Lattanzi, and Nikos Parotsidis. Fully dynamic k-clustering in $\tilde{O}(k)$ update time. *Advances in Neural Information Processing Systems*, 36, 2024.

Christopher M. Bishop. *Pattern recognition and machine learning, 5th Edition*. Information science and statistics. Springer, 2007. ISBN 9780387310732. URL https://www.worldcat.org/oclc/71008143.

Christopher M Bishop. Model-based machine learning. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20120222, 2013.

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. SIAM, 2011.

M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert systems with applications*, 40(1):200–210, 2013.

Ke Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

Davin Choo, Christoph Grunau, Julian Portmann, and Václav Rozhon. k-means++: few more steps yield constant approximation. In *International Conference on Machine Learning*, pages 1909–1917. PMLR, 2020.

Vincent Cohen-Addad, Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Improved approximations for euclidean k-means and k-median, via nested quasi-independent sets. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1621–1628, 2022.

Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A ptas for k-means clustering based on weak coresets. In *Proceedings of the twenty-third annual symposium on Computational geometry*, pages 11–18, 2007.

Fabrizio Grandoni, Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Rakesh Venkat. A refined approximation for euclidean k-means. *Information Processing Letters*, 176:106251, 2022.

Christoph Grunau, Ahmet Alper Özüdoğru, Václav Rozhoň, and Jakub Tětek. A nearly tight analysis of greedy k-means++. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1012–1070. SIAM, 2023.

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.

Ragesh Jaiswal and Nitin Garg. Analysis of k-means++ for separable data. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 591–602. Springer, 2012.

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.

Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1 + \epsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 332–341. IEEE, 2004.

Silvio Lattanzi and Christian Sohler. A better k-means++ algorithm via local search. In *International Conference on Machine Learning*, pages 3662–3671, 2019.

Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Information Processing Letters*, 120:40–43, 2017.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Meena Mahajan, Prajakta Nimbhorkar, and Kasturi Varadarajan. The planar k-means problem is np-hard. *Theoretical Computer Science*, 442:13–21, 2012.

Konstantin Makarychev, Aravind Reddy, and Liren Shan. Improved guarantees for k-means++ and k-means++ parallel. *Advances in Neural Information Processing Systems*, 33:16142–16152, 2020.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Moshe Shechner, Or Sheffet, and Uri Stemmer. Private k-means clustering with stability assumptions. In *International Conference on Artificial Intelligence and Statistics*, pages 2518–2528. PMLR, 2020.

Dennis Wei. A constant-factor bi-criteria approximation guarantee for k-means++. *Advances in neural information processing systems*, 29, 2016.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contribution.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper aims to seek an instance that separates the greedy k-means++ and k-means++. Thus, there are some assumptions for the instance, but the instance assumption captures the instance in reality.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main contribution of this paper is theoretical. All proofs and assumptions are explicitly described in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper includes the experiments. The experiment aims to verify the theoretical findings of the paper. All required information for reproducibility is provided in the main text and supplemental material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The submission includes the source code of the experiment in supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper includes the experiments. All settings of the experiment are explicitly described in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experiment result includes different values of parameters. The statistical result is shown in a figure.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experimental details are described in the Experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The submission respects the NeurIPS code of ethics. The submission is theoretical work and there is no ethics issue.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This is a theoretical work and there is no societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This is a theoretical submission, and there is no safeguard issue.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The submission does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The submission does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The submission does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The submission does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The submission does not use any LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Missing Proofs in Section 3

Proof of Lemma 1: To show the lemma for both cases when $\mathcal{A}$ is the greedy and k-means++, assume that the greedy samples ℓ candidate points per iteration, where $\ell \in \{1, 2, \dots, \ln k\}$. Clearly, it captures both the greedy that samples ℓ candidates per iteration and k-means++ that samples exactly one candidate per iteration. For an arbitrary $r \geq \delta$, let $\mathcal{A}(r)$ denote the subevent of $\mathcal{A}$ in which the farthest distance between any sampled candidate and its corresponding cluster center in greedy k-means++ is exactly r. Clearly, $\Pr[\mathcal{A}] = \int_{4b(1+\theta)\ln k}^{\infty} \Pr[\mathcal{A}(r)] \mathrm{d}r$.

We now analyze $\Pr[\mathcal{A}(r)]$. To this end, we partition $\mathcal{A}(r)$ into k subevents $\{\mathcal{A}^{(t)}(r)\}_{t\in[k]}$, based on the iteration in which a candidate point at distance r from its cluster center is first sampled. Specifically, $\mathcal{A}^{(t)}(r)$ denotes the subevent in which such a point is sampled for the first time in iteration t.

We next prove an upper bound on each $\Pr[\mathcal{A}^{(t)}(r)]$. By the definition of the event, all centers selected by the algorithm before iteration t, i.e., the points in S_{t-1}, must lie within a distance less than r from their respective cluster centers. Consider all points in $\bigcup_{i\in[k]} C_i(r)$. By the triangle inequality, the distance from any such point to any selected center in S_{t-1} is at most $d + 2r$. Therefore, the total connection cost of these points to S_{t-1} is at most

$$(d + 2r)^2 \cdot \frac{1}{b} \cdot e^{-r/b} \cdot n.$$

According to Observation 1, the total connection cost of all points to S_{t-1} is at least $2b^2 \cdot n$. Since $\ell \leq \ln k$, from union bounds, we have

$$\Pr[\mathcal{A}^{(t)}(r)] \leq \ln k \cdot \frac{(d + 2r)^2 \cdot e^{-r/b}}{2b^3} .$$

Therefore,

$$\begin{aligned}
\Pr[\mathcal{A}] &= \int_{4b(1+\theta)\ln k}^{\infty} \Pr[\mathcal{A}(r)]\, \mathrm{d}r \\
&\leq \int_{4b(1+\theta)\ln k}^{\infty} k \ln k \cdot \frac{(d + 2r)^2 \cdot e^{-r/b}}{2b^3}\, \mathrm{d}r \\
&\qquad\qquad\qquad \text{(Sum of the upper bounds over the } k \text{ subevents)} \\
&\leq \int_{4b(1+\theta)\ln k}^{\infty} k^{1+2\theta} \ln k \cdot \frac{16r^2 \cdot e^{-r/b}}{b}\, \mathrm{d}r \\
&\leq 16 \cdot k^{1+2\theta} \ln k \cdot (2b + 4b(1+\theta)\ln k)^2 \cdot e^{-4(1+\theta)\ln k} ,
\end{aligned}$$

which is $o(1/k)$ asymptotically. Similarly, for the contribution of event $\mathcal{A}$ to the expected objective, we have

$$\begin{aligned}
\Pr[\mathcal{A}] \cdot \mathbb{E}[\mathsf{OBJ} \mid \mathcal{A}] &= \int_{4b(1+\theta)\ln k}^{\infty} \Pr[\mathcal{A}(r)] \cdot \mathbb{E}[\mathsf{OBJ} \mid \mathcal{A}(r)]\, \mathrm{d}r \\
&\leq \int_{4b(1+\theta)\ln k}^{\infty} \Pr[\mathcal{A}(r)] \cdot n \cdot (2b^2 + (d + 2r)^2)\, \mathrm{d}r \qquad \text{(Observation 2)} \\
&\leq \int_{4b(1+\theta)\ln k}^{\infty} n \cdot (2b^2 + (d + r)^2) \cdot k \ln k \cdot \frac{(d + r)^2 \cdot e^{-r/b}}{2b^3}\, \mathrm{d}r ,
\end{aligned}$$

which is bounded by $o(n/k)$ asymptotically. ■

Proof of Lemma 2: The first argument (2a) follows directly. By the concentration assumption, for any uncovered cluster, the distance from its center μ_i to solution S is $\Omega(k^\theta)$, which yields a connection cost of $\Omega(k^{2\theta} \cdot |C_i|)$ by Observation 2. Similarly, since the distance from μ_i to solution S is $O(\ln k)$ for covered clusters, the first part of the second argument follows. It remains to show the second part.

Consider the first iteration t in which the algorithm covers cluster C_i. In this iteration, the algorithm samples a set of candidate points from $\sigma(C_i)$. Suppose that p such points are sampled, denoted by $P = \{x_1, \ldots, x_p\}$, where $1 \le p \le \ln k$. We will show that the expected maximum connection cost from the cluster center μ_i to the sampled candidates is $O((\ln p)^2)$. Then, by Observation 2, the expected total connection cost for cluster C_i is also bounded by $O((\ln p)^2)$. Since $p \le \ln k$, this implies a bound of $O((\ln \ln k)^2)$.

As t is the first iteration in which C_i is covered, all centers selected before this iteration, i.e., the points in S_{t-1}, are far from the points in $\sigma(C_i)$, with distances lying in the range $[d - 2\delta,\, d + 2\delta]$. Recalling that $d = \Theta(k^\theta)$ and $\delta = \Theta(\ln k)$, we have for any point $x \in \sigma(C_i)$, the connection cost $\varphi(x, S_{t-1})$ differs from that of other points in $\sigma(C_i)$ by at most a small constant factor, especially when k is large. Consequently, up to a constant-factor loss in expectation, we may assume that the algorithm samples p points uniformly at random from $\sigma(C_i)$. For notational simplicity, we omit the conditioning on the event that exactly p candidates are sampled from $\sigma(C_i)$ in the expectation below. We have

$$\begin{aligned}
\mathbb{E}\left[\max_{x\in P} \|x - \mu_i\|_2^2\right] &\le \int_0^\infty \Pr\left[\max_{x\in P} \|x - \mu_i\|_2^2 \ge r\right] \mathrm{d}r \\
&= \int_0^\infty \Pr\left[\max_{x\in P} \|x - \mu_i\|_2 \ge \sqrt{r}\right] \mathrm{d}r \\
&= \int_0^\infty 2h \cdot \Pr\left[\max_{x\in P} \|x - \mu_i\|_2 \ge h\right] \mathrm{d}h \\
&= \int_0^\infty 2h \cdot \Pr\left[\exists\, x \in P \text{ such that } \|x - \mu_i\|_2 \ge h\right] \mathrm{d}h \\
&= \int_0^\infty 2h \cdot \left(1 - (1 - e^{-h/b})^p\right) \mathrm{d}h\ .
\end{aligned}$$

A standard calculation can show that the expression above is asymptotically bounded by $O((\ln p)^2)$. From the fact of the exponential distribution, it is straightforward to see that $\mathbb{E}\left[\max_{x\in P} \|x - \mu_i\|_2^2\right] = \Omega(1)$. Thus, we have established the upper and lower bounds on C_i's expected connected cost when covered as claimed. This completes the proof of (2b). ■

Proof of Claim 1: We shall use the following concentration bound:

Theorem 4 (Hoeffding's inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be independent random variables such that $a_i \le X_i \le b_i$. Let $X = \sum_{i\in[n]} X_i$ and $\mu = \mathbb{E}[X]$. For all $t > 0$, we have*

$$\Pr[X - \mu \ge t] \le \exp\Big(-\frac{2t^2}{\sum_{i\in[n]}(b_i - a_i)^2}\Big)$$

Let random variable Y_t denote the connection cost of a cluster covered in the t-th iteration with $t \in \{1, 2, \ldots, t_0\}$. By (2b) of Lemma 2, we know that $Y_t \le \rho_1 \cdot (\ln k)^2 \cdot \frac{n}{k}$ and $\mathbb{E}[\sum_{t\in[t_0]} Y_t] \le \rho_2 \cdot (\ln \ln k)^2 \cdot \frac{n}{k} \cdot t_0$ where we are allowed to let $\rho_1 = (4b(1+\theta))^2$, and $\rho_2 = b^2$. For simplicity, we denote $\mathbb{E}[\sum_{t\in[t_0]} Y_t]$ by μ and its upper bound $\rho_2 \cdot (\ln\ln k)^2 \cdot \frac{n}{k} \cdot t_0$ by C. Applying Chernof-Hoeffding's inequality (Theorem 4 in the appendix), we have

$$\begin{aligned}
\Pr\Big[\sum_{t\in[t_0]} Y_t \ge 2\cdot C\Big] \le \Pr\Big[\sum_{t\in[t_0]} Y_t \ge \mu + C\Big] &\le \exp\Big(-\frac{2C^2}{\sum_{t\in[t_0]}(\rho_1 \cdot (\ln k)^2 \cdot \frac{n}{k})^2}\Big) \\
&\le \exp\Big(-2k \cdot \big(\frac{\rho_2 \ln\ln k}{\rho_1 \ln k}\big)^2\Big)
\end{aligned}$$

■

B Missing Proofs in Section 4

Proof of Lemma 4: This proof follows a similar analysis to that in Lemma 2. By Observation 2, the total connection cost of a cluster C_i is $(2b^2 + h^2) \cdot |C_i|$, where h denotes the distance from the cluster center μ_i to the current set of selected centers S. When the cluster is not yet covered, $h \in [d - \delta,\, d + 2\delta]$, which is of order $\Theta(k^\theta)$; whereas once the cluster is covered, b is still constant that implies the total connection cost is $\Omega(|C_i|)$. This completes the proof of the lemma. ■

C Distributions with Sub-Exponential Tails

In the main body, we present the results and detailed proofs under the assumption that each cluster follows an exponential distribution. However, this assumption is not essential—in fact, the same analysis applies to any subexponential-tailed distributions with a mild assumption. In an extreme case, all points in a cluster share the same location as the center. k-means++ or the greedy can always pick a new uncovered cluster in every iteration. To avoid points being overly concentrated around the centers, we assume that the distributions satisfy $\Pr_{x \sim \mathcal{D}}[x \geq \epsilon] = \Omega(1)$. Clearly, the exponential distribution has this property. Below, we provide three representative examples: Gaussian, sub-Gaussian, and sub-exponential distributions. Each of them exhibits tail probabilities that decay exponentially with distance from the center.

Gaussian Distribution

- *Parameters:* Mean vector $b \in \mathbb{R}$, variance σ^2.
- *Tail bound:* For $X \sim \mathcal{N}(b, \sigma^2)$,

$$\Pr(X - b \geq r) \leq \exp\left(-\frac{r^2}{2\sigma^2}\right),$$

  which decays exponentially in r^2.

Sub-Gaussian Distribution

- *Parameter:* Mean $b \in \mathbb{R}$, variance proxy parameter σ^2.
- *Tail bound:* For a sub-Gaussian random vector X,

$$\Pr(X - b \geq r) \leq \exp\left(-\frac{cr^2}{\sigma^2}\right)$$

  for some absolute constant $c > 0$, showing exponential decay in r^2.

Sub-Exponential Distribution

- *Parameters:* Mean $b \in \mathbb{R}$, sub-exponential parameters (ν, α), where ν^2 reflects the variance-like behavior, and α controls the tail heaviness.
- *Tail bound:* For a sub-exponential random variable X,

$$\Pr(X - b \geq r) \leq \begin{cases} \exp\left(-\frac{r^2}{2\nu^2}\right), & \text{for } 0 \leq r \leq \frac{\nu^2}{\alpha}, \\ \exp\left(-\frac{r}{2\alpha}\right), & \text{for } r > \frac{\nu^2}{\alpha}, \end{cases}$$

  showing sub-Gaussian decay for small t and exponential decay for large t.

The sub-exponential distribution is a well-known generalization of the sub-Gaussian and the Gaussian distribution. For completeness, we will show the results in the main body under the sub-exponential distributions with a mild assumption. First, we restate our assumption. We denote the sub-exponential distribution by $\mathcal{D} = \mathsf{subE}(\nu, \alpha)$. Each cluster has a different distribution $\mathcal{D}_i = \mathsf{subE}(\nu_i, \alpha_i)$ with a mean b_i. Let $\lambda \geq 1$ be a universal constant which is the upper bound of all constant parameters. The input point set X satisfying the three properties:

- *Sub-Exponentially Distributed:* For each cluster C_i, the density of points decreases sub-exponentially as the distance from the center increases. Specifically,

$$\frac{|C_i(r)|}{|C_i|} \sim \mathsf{subE}(\nu_i, \alpha_i)$$

- *Well Separable*: The minimum distance d between any two centers of the optimal clusters is sufficiently large, i.e., $d = k^\theta \cdot \lambda$, for a constant $\theta > 0$.
- *Well Spread*: The optimal clusters are roughly homogeneous, with the number of points in each cluster and the distances between clusters differing by at most a constant factor. For the cluster C_i, we denote the number of points by n_i. Further, we have

$$\Pr_{x \sim \mathcal{D}_i}[x \geq \epsilon_i] \geq p_i \quad \forall i \in [k]$$

where ϵ_0, p_0 are some constant.

The universal constant controls the effects from the heterogeneous parameters where $1/\lambda \leq \nu_i, \alpha_i, b_i, n_i/(n/k), \epsilon_i, p_i \leq \lambda$.

Theorem 5 (Corresponding to Theorem 1)**.** *Given any general EWW point set X, the greedy k-means++ algorithm admits an expected approximation ratio of $O((\log\log k)^2)$.*

Similarly, we define the concentration balls as follows.

Definition 2 (Corresponding to Definition 1)**.** *We define the* concentration radius $\delta := 4\lambda(1+\theta)\ln k$*, where θ and b are input parameters. For each cluster C_i with center μ_i, we define its* concentration ball $\sigma(C_i)$ *as the set of all points in C_i whose distance to μ_i is at most δ.*

Lemma 7 (Corresponding to Lemma 1)**.** *Let $\mathcal{A}$ denote the event that greedy k-means++ samples at least one candidate point outside the concentration balls during any iteration. Given any general EWW point set X, the probability that $\mathcal{A}$ occurs is at most $1/k$. Furthermore, the contribution of this event to the expected objective can be bounded:*

$$\Pr[\mathcal{A}] \cdot \mathbb{E}[\mathsf{OBJ} \mid \mathcal{A}] \leq \frac{n}{k},$$

where $\mathbb{E}[\mathsf{OBJ} \mid \mathcal{A}]$ denotes the expected objective value (i.e., total connection cost) conditioned on the occurrence of $\mathcal{A}$.

Proof. To show the lemma for both cases when $\mathcal{A}$ is the greedy and k-means++, assume that the greedy samples ℓ candidate points per iteration where $\ell \in [1, \log k]$. Clearly, it captures both the greedy that samples ℓ candidates per iteration and k-means++ that samples exactly one candidate per iteration. For an arbitrary $r \geq \delta$, let $\mathcal{A}(r)$ denote the subevent of $\mathcal{A}$ in which the farthest distance between any sampled candidate and its corresponding cluster center in greedy k-means++ is exactly r. Clearly, $\Pr[\mathcal{A}] = \int_{4\lambda(1+\theta)\ln k}^{\infty} \Pr[\mathcal{A}(r)]\mathrm{d}r$.

We now analyze $\Pr[\mathcal{A}(r)]$. To this end, we partition $\mathcal{A}(r)$ into k subevents $\{\mathcal{A}^{(t)}(r)\}_{t\in[k]}$, based on the iteration in which a candidate point at distance r from its cluster center is first sampled. Specifically, $\mathcal{A}^{(t)}(r)$ denotes the subevent in which such a point is sampled for the first time in iteration t.

We next prove an upper bound on each $\Pr[\mathcal{A}^{(t)}(r)]$. By the definition of the event, all centers selected by the algorithm before iteration t, i.e., the points in S_{t-1}, must lie within a distance less than r from their respective cluster centers. Consider all points in $\bigcup_{i\in[k]} C_i(r)$. By the triangle inequality, the distance from any such point to any selected center in S_{t-1} is at most $d + 2r$. Therefore, the total connection cost of these points to S_{t-1} for a cluster i is at most

$$(d+2r)^2 \cdot e^{-r/(2\alpha_i)} \cdot n_i \leq (d+2r)^2 \cdot e^{-r/(2\lambda)} \cdot \lambda n/k.$$

From the well-spread property of the dataset, the total connection cost of all points to S_{t-1} for a cluster i is at least $\epsilon_i^2 p_i \cdot n_i \geq (n/k)/\lambda^4$. Since $\ell \leq \log k$, from union bounds, we have

$$\Pr[\mathcal{A}^{(t)}(r)] \leq \log k \cdot \frac{(d+2r)^2 \cdot e^{-r/(2\lambda)} \cdot \lambda}{1/\lambda^4} .$$

Therefore,

$$\Pr[\mathcal{A}] = \int_{4\lambda(1+\theta)\ln k}^{\infty} \Pr[\mathcal{A}(r)]\,\mathrm{d}r$$

$$
\begin{aligned}
&\le \int_{4\lambda(1+\theta)\ln k}^{\infty} k\log k\cdot\lambda^5(d+2r)^2\cdot e^{-r/(2\lambda)}\,\mathrm{d}r \\
&\qquad\qquad\qquad\qquad\qquad\text{(Sum of the upper bounds over the } k \text{ subevents)} \\
&\le \int_{4\lambda(1+\theta)\ln k}^{\infty} 16\lambda^7 k^{1+2\theta}\log k\cdot r^2\cdot e^{-r/(2\lambda)}\,\mathrm{d}r \\
&\le 128\cdot\lambda^{10}k^{1+2\theta}\log k\cdot(2\lambda+4\lambda(1+\theta)\ln k)^2\cdot e^{-4\lambda(1+\theta)\ln k}\ ,
\end{aligned}
$$

which is $o(1/k)$ asymptotically. Similarly, for the contribution of event $\mathcal{A}$ to the expected objective, we have

$$
\begin{aligned}
\Pr[\mathcal{A}]\cdot\mathbb{E}[\mathsf{OBJ}\mid\mathcal{A}] &= \int_{4\lambda(1+\theta)\ln k}^{\infty}\Pr[\mathcal{A}(r)]\cdot\mathbb{E}[\mathsf{OBJ}\mid\mathcal{A}(r)]\,\mathrm{d}r \\
&\le \int_{4\lambda(1+\theta)\ln k}^{\infty}\Pr[\mathcal{A}(r)]\cdot n\cdot(2\lambda^2+(d+2r)^2)\,\mathrm{d}r \qquad \text{(Observation 2)} \\
&\le \int_{4\lambda(1+\theta)\ln k}^{\infty} n\cdot(2\lambda^2+(d+r)^2)\cdot k\log k\cdot\lambda^5(d+2r)^2\cdot e^{-r/(2\lambda)}\,\mathrm{d}r\ ,
\end{aligned}
$$

which is bounded by $o(n/k)$ asymptotically. □

Lemma 8 (Corresponding to Lemma 2)**.** *Given any general EWW point set X, under the concentration assumption, for each cluster C_i , we have:*

(8a) *If greedy k-means++ does not cover this cluster, then its total connection cost is $\Omega(k^{2\theta}\cdot|C_i|)$.*

(8b) *If greedy k-means++ covers this cluster, then the total connection cost for C_i does not exceed $O((\ln k)^2\cdot|C_i|)$, and further, its expectation is $O((\ln\ln k)^2\cdot|C_i|)$ and $\Omega(|C_i|)$.*

Proof. The first argument (8a) follows directly. By the concentration assumption, for any uncovered cluster, the distance from its center μ_i to solution S is $\Omega(k^\theta)$, which yields a connection cost of $\Omega(k^{2\theta}\cdot|C_i|)$ by Observation 2. Similarly, since the distance from μ_i to solution S is $O(\ln k)$ for covered clusters, the first part of the second argument follows. It remains to show the second part.

We apply the same strategy in Lemma 2. As t is the first iteration in which C_i is covered, all centers selected before this iteration, i.e., the points in S_{t-1}, are far from the points in $\sigma(C_i)$, with distances lying in the range $[d-2\delta,\ d+2\delta]$. Recalling that $d=\Theta(k^\theta)$ and $\delta=4\lambda(1+\theta)\ln k$, we have for any point $x\in\sigma(C_i)$, the connection cost $\varphi(x,S_{t-1})$ differs from that of other points in $\sigma(C_i)$ by at most a small constant factor, especially when k is large. Consequently, up to a constant-factor loss in expectation, we may assume that the algorithm samples p points uniformly at random from $\sigma(C_i)$. For notational simplicity, we omit the conditioning on the event that exactly p candidates are sampled from $\sigma(C_i)$ in the expectation below. We have

$$
\begin{aligned}
\mathbb{E}\left[\max_{x\in P}\|x-\mu_i\|_2^2\right] &\le \int_0^\infty \Pr\left[\max_{x\in P}\|x-\mu_i\|_2^2\ge r\right]\,\mathrm{d}r \\
&= \int_0^\infty \Pr\left[\max_{x\in P}\|x-\mu_i\|_2\ge\sqrt{r}\right]\,\mathrm{d}r \\
&\le \int_0^{(\lambda\ln p)^2} 1\mathrm{d}h + \int_{(\lambda\ln p)^2}^{\infty} 2h\cdot\Pr\left[\max_{x\in P}\|x-\mu_i\|_2\ge h\right]\mathrm{d}h \\
&= (\lambda\ln p)^2 + \int_{(\lambda\ln p)^2}^{\infty} 2h\cdot\Pr\left[\exists\, x\in P \text{ such that } \|x-\mu_i\|_2\ge h\right]\mathrm{d}h \\
&\le (\lambda\ln p)^2 + \int_{(\lambda\ln p)^2}^{\infty} 2h\cdot\ln p\cdot e^{-h/(2\lambda)}\mathrm{d}h
\end{aligned}
$$

A standard calculation can show that the expression above is asymptotically bounded by $O((\ln p)^2)$. From the well-spread property, it is straightforward to see that $\mathbb{E}\left[\max_{x\in P}\|x-\mu_i\|_2^2\right]=\Omega(1)$. Thus, we have established the upper and lower bounds on C_i's expected connected cost when covered as claimed. This completes the proof. □

Lemma 9 (Corresponding to Lemma 3)**.** *Given any general EWW point set X, in each iteration $t \le k - O(k^{1-2\theta} \cdot (\log\log k)^2)$, the greedy k-means++ algorithm covers a new cluster with probability at least $1 - 1/k^{2\theta+2}$.*

Proof. Since we have Lemma 8 for sub-exponential distributions, using the same proof for Lemma 2, we could have the lemma. For the sake of completeness, we provide the following proof with a slightly different constant factor. We first leverage (8a) of Lemma 8 to show that, with high probability, the greedy k-means++ algorithm covers a new cluster in each iteration up to iteration $k - O(k^{1-2\theta} \cdot (\ln k)^2)$. Then, by applying (8b) of Lemma 8 along with the Chernoff bound, we demonstrate that the algorithm continues to cover a new cluster in each iteration, with high probability, even during the range of iterations from $k - O(k^{1-2\theta} \cdot (\ln k)^2)$ to $k - O(k^{1-2\theta} \cdot (\ln\ln k)^2)$.

First, consider an iteration $k \le t_0 := k - 2^{2\theta+5} \cdot k^{1-2\theta} \cdot (\ln k)^2$. Lemma 8 implies that the greedy comparison step always prefers points from uncovered concentration balls over those from already covered ones. This is because if the greedy selects a point from an uncovered concentration ball (by (8a)), it decreases the objective by $\Omega(n/k)k^{2\theta}$ while if it selects from a covered concentration ball, it only decreases the objective by $O(n/k)(\ln k)^2$ (by (8b)). Thus, it follows that, in iteration t, if at least one of the $\ln k$ sampled candidates is from an uncovered concentration ball, then the algorithm is guaranteed to cover a new cluster in that iteration. Suppose that p clusters have already been covered by S_{t-1} at the beginning of iteration t (so $p < t$). By (8b) of Lemma 8 and the well-spread property, the total connection cost contributed by the covered clusters is at most $\lambda\cdot\frac{n}{k}\cdot p\cdot(4\lambda(1+\theta)\ln k)^2$, while the total connection cost from the uncovered clusters is at least $1/\lambda \cdot \frac{n}{k} \cdot (k-p) \cdot \lambda^2 k^{2\theta}$, Therefore, the probability[5] that the algorithm fails to sample any point from the uncovered concentration balls is at most

$$\left(\frac{\lambda p \cdot (4\lambda(1+\theta)\ln k)^2}{\lambda p \cdot (4\lambda(1+\theta)\ln k)^2 + (k-p)\cdot \lambda k^{2\theta}}\right)^{\ln k} \le 1/k^{2\theta+5} \text{ when } p \le t_0.$$

Next, we analyze the time period between iteration t_0 and $t_1 := k - 2^{2\theta+5} \cdot k^{1-2\theta} \cdot (\ln\ln k)^2$. From the earlier analysis and by applying a union bound, we know that with probability at least $1 - 1/k^{2\theta+4}$, the algorithm has already covered t_0 clusters within the first t_0 iterations. Then, according to Lemma 8, conditioned on this event, the expected connection cost of these covered clusters is lower bounded by $\Omega(\frac{n}{k}\cdot t_0)$, which is asymptotically much larger than the upper bound on the connection cost for any single cluster, $O\left(\frac{n}{k}\cdot(\ln k)^2\right)$. This suggests the connection cost of the covered clusters is well-concentrated.

We shall upper bound the total connection cost of the covered clusters in the first t_0 iterations via the concentration bound. Note that each cluster can be approximately treated as being sampled uniformly based on the analysis in the proof of Lemma 8, which enables us to use concentration inequalities.

Let random variable Y_t denote the connection cost of a cluster covered in the t-th iteration with $t \in \{1, 2, \ldots, t_0\}$. By (8b) of Lemma 8, we know that $Y_t \le \rho_1 \cdot (\ln k)^2 \cdot \frac{n}{k}$ and $\mathbb{E}[\sum_{t\in[t_0]} Y_t] \le \rho_2 \cdot (\ln\ln k)^2 \cdot \frac{n}{k} \cdot t_0$ where we are allowed to let $\rho_1 = \lambda(4\lambda(1+\theta))^2$, and $\rho_2 = 4\lambda^3$. For simplicity, we denote $\mathbb{E}[\sum_{t\in[t_0]} Y_t]$ by μ and its upper bound $\rho_2 \cdot (\ln\ln k)^2 \cdot \frac{n}{k} \cdot t_0$ by C. Applying Chernof-Hoeffding's inequality (Theorem 4 in the appendix), we have

$$\begin{aligned}\Pr\Big[\sum_{t\in[t_0]} Y_t \ge 2\cdot C\Big] \le \Pr\Big[\sum_{t\in[t_0]} Y_t \ge \mu + C\Big] &\le \exp\Big(-\frac{2C^2}{\sum_{t\in[t_0]}(\rho_1\cdot(\ln k)^2\cdot\frac{n}{k})^2}\Big)\\ &\le \exp\Big(-2k\cdot(\frac{\rho_2\ln\ln k}{\rho_1\ln k})^2\Big)\end{aligned}$$

Thus, with probability at least $1 - \exp\left(\Theta(1)\cdot(-k)\cdot\left(\frac{\ln\ln k}{\ln k}\right)^2\right)$, the total connection cost of the clusters covered in the first t_0 iterations is at most $4\lambda^3 \cdot \frac{n}{k} \cdot t_0 \cdot (\ln\ln k)^2$.

Consider any iteration $t \in (t_0, t_1]$. The algorithm can cover at most $t_1 - t_0$ new clusters during the interval from iteration t_0 to t_1, and each newly covered cluster contributes at most $O\left(\frac{n}{k}\cdot(\ln k)^2\right)$ to

[5] As mentioned earlier, for simplicity, we assume that each cluster is of equal size and that they are equidistant from one another. We note that if the clusters are not exactly equal in size but differ by only a constant factor, the probability will increase by at most a constant, and the overall order of magnitude will remain the same.

the connection cost. Thus, with a probability of at least

$$\left(1-\frac{1}{k^{2\theta+4}}\right)\left(1-\exp\left(\Theta(1)\cdot(-k)\cdot\left(\frac{\ln\ln k}{\ln k}\right)^2\right)\right) ,$$

the total connection cost of the already covered clusters at the beginning of iteration t is at most

$$A := 16\lambda^3(1+\theta)^2\cdot\frac{n}{k}\cdot\left(b^2\cdot t_0\cdot(\ln\ln k)^2+(t_1-t_0)\cdot(\ln k)^2\right) .$$

Note that the total connection cost from the uncovered clusters is at least $B := \lambda\frac{n}{k}\cdot(k-t_0)\cdot k^{2\theta}$. Similar to the analysis in the previous case, the probability that the algorithm fails to sample any point from the uncovered concentration balls is at most $(A/(A+B))^{\ln k}$. This implies that the probability that greedy k-means++ fails to cover a new cluster is at most $1/k^{2\theta+2}$. □

Proof of Theorem 5. By Lemma 9 and a union bound, we have that, with probability at least $1-1/k^{2\theta+1}$, the greedy k-means++ algorithm covers $k-O\left(k^{1-2\theta}\cdot(\ln\ln k)^2\right)$ clusters and achieves an approximation ratio of $O((\ln\ln k)^2)$. If this case does not occur, we can simply upper bound the objective by $O(n\cdot k^{2\theta})$. Taking expectation over both cases, we obtain an expected approximation ratio of $O((\ln\ln k)^2)$. □

Theorem 6 (Corresponding to Theorem 2)**.** *Given any general EWW point set X with parameter $\theta\in(0,1/2]$, the k-means++ algorithm admits an expected approximation ratio of $\Omega(\log k)$.*

Lemma 10 (Corresponding to Lemma 4)**.** *Given any general EWW point set X, under the concentration assumption, for each cluster C_i , we have:*

- *If k-means++ does not cover this cluster, then its total connection cost is $\Theta(k^{2\theta}\cdot|C_i|)$.*
- *If k-means++ covers this cluster using exactly one center—that is, the final solution includes exactly one point from $\sigma(C_i)$—then the total connection cost for C_i is $\Omega(|C_i|)$.*

Proof. By Observation 2, the total connection cost of a cluster C_i is $(2\lambda^2+h^2)\cdot|C_i|$, where h denotes the distance from the cluster center μ_i to the current set of selected centers S. When the cluster is not yet covered, $h\in[d-\delta,\ d+2\delta]$, here $d=\lambda k^\theta$, which is of order $\Theta(k^\theta)$; whereas once the cluster is covered, the well-spread property, where $\Pr_{x\sim\mathcal{D}_i}[x\geq\epsilon_i]\geq p_i\ \forall i\in[k]$, implies the total connection cost is at least $|C_i|/\lambda^2=\Omega(|C_i|)$. This completes the proof of the lemma. □

Proof of Theorem 6. Lemma 10 shows that if the number of uncovered clusters is p, then the objective value is at least $\lambda\frac{n}{k}\cdot p\cdot k^{2\theta}$, omitting constant factors for simplicity. Therefore, to establish a lower bound of $\Omega(\ln k)$ on the approximation ratio, it suffices to show that, in expectation, k-means++ leaves $\Omega(\ln k\cdot k^{1-2\theta})$ clusters uncovered. This also explains why we require $\theta\in(0,1/2]$: otherwise, $\ln k\cdot k^{1-2\theta}$ would be subconstant, rendering the argument meaningless.

We partition all possible outcomes of the algorithm into two cases based on the number of uncovered clusters: (1) the final number of uncovered clusters is at least Δ, and (2) the final number of uncovered clusters is less than Δ, where $\Delta=\ln k\cdot k^{1-2\theta}$. Clearly, in all outcomes falling into the first case, the approximation ratio is $\Omega(\ln k)$. Next, we analyze the second case and show that, in expectation, the number of uncovered clusters remains $\Omega(\Delta)$.

Consider an arbitrary iteration t. In the second case, where the final number of uncovered clusters is less than Δ, the number of clusters already covered by the solution S_{t-1} at the beginning of this iteration must be at least $t-\Delta$. Otherwise, even if every subsequent iteration covers a new cluster, the final number of uncovered clusters would exceed Δ, contradicting the assumption. Then, by the pigeonhole principle, at least $t-2\Delta$ clusters must be covered by exactly one center—that is, S_{t-1} contains exactly one point from each of these clusters. By Lemma 10 and the well-spread property, the total connection cost of the covered clusters is at least $1/\lambda^3\cdot\frac{n}{k}\cdot(t-2\Delta)$, while the total connection cost of the uncovered clusters is at most $\lambda^3\cdot\frac{n}{k}\cdot(k-t+\Delta)\cdot k^{2\theta}$. Therefore, in each iteration $t>2\Delta$, the probability that the k-means++ algorithm fails to cover a new cluster is at least $1/\lambda^6\cdot\frac{t-2\Delta}{(t-2\Delta)+(k-t+\Delta)\cdot k^{2\theta}}$.

We compute the expected number of uncovered clusters by summing the failure probabilities across all iterations (conditioned on the second case). Specifically, we have:

$$\mathbb{E}[\text{number of uncovered clusters}] \geq 1/\lambda^6 \cdot \sum_{t>2\Delta}^{k} \frac{t-2\Delta}{(t-2\Delta)+(k-t+\Delta)\cdot k^{2\theta}}$$

$$\geq 1/\lambda^6 \cdot \sum_{t\geq k/2}^{k} \frac{t-2\Delta}{(t-2\Delta)+(k-t+\Delta)\cdot k^{2\theta}} \geq 1/\lambda^6 \cdot \sum_{t\geq k/2}^{k} \frac{k/4}{k/4+(k-t+\Delta)\cdot k^{2\theta}} \qquad (\Delta = o(k))$$

$$= 1/\lambda^6 \cdot k^{1-2\theta} \cdot \sum_{t\geq k/2}^{k} \frac{1}{k^{-2\theta}+4(k-t+\Delta)} \geq 1/\lambda^6 \cdot k^{1-2\theta} \ln\left(\frac{k/2+\Delta}{\Delta}\right)$$

$$\geq 1/\lambda^6 \cdot k^{1-2\theta} \ln\left(\frac{k^{2\theta}}{2\ln k}\right) = \Omega(k^{1-2\theta}\ln k) \ ,$$

which implies that the expected number of uncovered clusters is $\Omega(\Delta)$ and completes the proof. □

Theorem 7 (Corresponding to Theorem 3)**.** *Given any general EWW point set with parameter $\theta > 1/2$, the probability that greedy k-means++ covers all optimal clusters is greater than that of k-means++.*

Let event $\mathcal{B}$ denote the bad event that the algorithm selects a point from an already covered cluster.

Lemma 11 (Corresponding to Lemma 5)**.** *Given any general EWW point set X, the probability that k-means++ encounters $\mathcal{B}$ is at least $\frac{k-1}{k-1+k^{2\theta}}$.*

Proof. We partition the bad event into two sub-events based on the time at which $\mathcal{B}$ first occurs: (1) k-means++ encounters $\mathcal{B}$ before the last iteration k, and (2) k-means++ first encounters $\mathcal{B}$ at the last iteration k. We denote these two sub-events as $\mathcal{P}$ and $\mathcal{Q}$, respectively. By expanding the conditional probability of the second sub-event, we derive a lower bound on the probability that k-means++ encounters $\mathcal{B}$:

$$\Pr[k\text{-means++ encounters } \mathcal{B}] = \Pr[\mathcal{P}] + \Pr[\mathcal{Q}] = \Pr[\mathcal{P}] + \Pr[\neg\mathcal{P}] \cdot \Pr[\mathcal{Q} \mid \neg\mathcal{P}] \geq \Pr[\mathcal{Q} \mid \neg\mathcal{P}].$$

Conditioned on $\neg\mathcal{P}$, the notion of "first" in event $\mathcal{Q}$ is not essential—$\Pr[\mathcal{Q} \mid \neg\mathcal{P}]$ simply equals the probability that k-means++ samples a point from one of the $k-1$ already covered clusters. By Lemma 10, the total connection cost of the already covered clusters is at least $\frac{n}{k} \cdot (k-1)/\lambda^3$, while the total connection cost of the last uncovered cluster is at most $\frac{n}{k} \cdot \lambda^3 k^{2\theta}$. Thus, the probability that k-means++ encounters event $\mathcal{B}$ can be lower bounded by $1/\lambda^6 \cdot \frac{k-1}{k-1+k^{2\theta}}$. □

Lemma 12 (Corresponding to Lemma 6)**.** *Given any general EWW point set X, the probability that greedy k-means++ encounters $\mathcal{B}$ is at most $\left(\frac{16e(1+\theta)^2\cdot(k-1)\cdot(\log k)^2}{(k-1)\cdot(\log k)^2+k^{2\theta}}\right)^{\log k}$.*

Proof. To upper bound the probability for greedy k-means++, we partition the bad event into k sub-events $\{\mathcal{P}_t\}_{t\in[k]}$, where $\mathcal{P}_t$ denotes the event that greedy k-means++ encounters $\mathcal{B}$ for the first time in iteration t. Similarly, we then expand the probability of $\mathcal{P}_t$ using conditional probability:

$$\Pr[\text{greedy } k\text{-means++ encounters } \mathcal{B}] = \sum_{t\in[k]} \Pr[\mathcal{P}_t]$$

$$= \sum_{t\in[k]} \Pr[\neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{t-1})] \cdot \Pr[\mathcal{P}_t \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{t-1})] \leq k \cdot \Pr[\mathcal{P}_k \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{k-1})]$$

where the last inequality uses the fact that $\Pr[\mathcal{P}_t \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{t-1})]$ attains its maximum at $t = k$.

The term $\Pr[\mathcal{P}_k \mid \neg(\mathcal{P}_1 \vee \cdots \vee \mathcal{P}_{k-1})]$ simply equals the probability that greedy k-means++ samples all $\ln k$ candidates from one of the $k-1$ already uncovered clusters. By Lemma 8, the total connection cost of the already covered cluster is at most $\lambda \cdot \frac{n}{k} \cdot (k-1) \cdot (4\lambda(1+\theta)\ln k)^2$, while the total connection cost of the last uncovered cluster is at most $\lambda^3 \cdot \frac{n}{k} \cdot k^{2\theta}$, omitting constant factors for simplicity. Thus, the probability that greedy k-means++ encounters $\mathcal{B}$ can be upper bounded by

$$k \cdot \left(\frac{(4(1+\theta))^2 (k-1) \cdot (\ln k)^2}{(k-1) \cdot (\ln k)^2 + k^{2\theta}} \right)^{\ln k} = \left(\frac{16e(1+\theta)^2 \cdot (k-1) \cdot (\ln k)^2}{(k-1) \cdot (\ln k)^2 + k^{2\theta}} \right)^{\ln k} .$$

□

Proof of Theorem 7. Lemma 11 and Lemma 12, through a series of mathematical calculations, directly establish the theorem. More specifically, when $\theta > 1/2$, the lower bound on the failure probability for k-means++ is $\Theta(k^{1-2\theta})$, while the upper bound for greedy k-means++ is $\Theta\left(k^{16e \ln \ln k + (1-2\theta) \ln k}\right)$. As the former asymptotically dominates the latter, we can conclude that greedy k-means++ has a higher probability of covering all optimal clusters than k-means++. □

D Additional Experiments

In this section, we present more experimental results on varying k, which validates our theory works for different k.

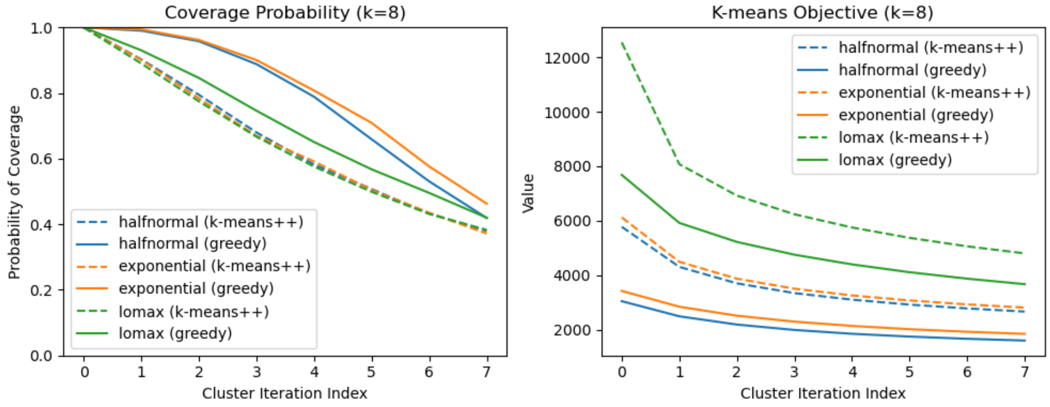

Figure 2: Coverage probability and k-means objective over iterations for $k = 8$

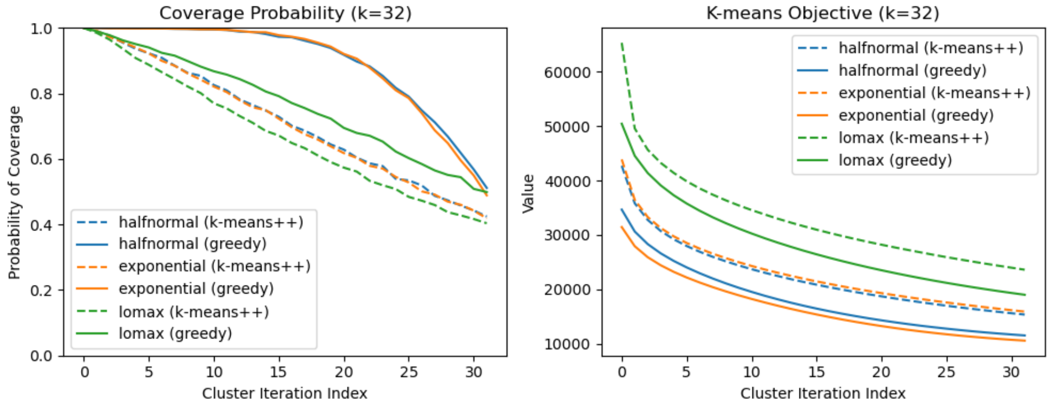

Figure 3: Coverage probability and k-means objective over iterations for $k = 32$

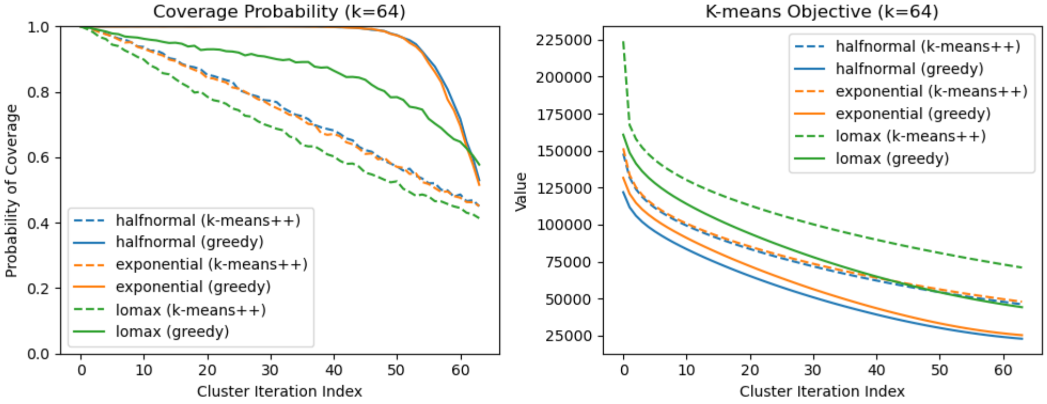

Figure 4: Coverage probability and k-means objective over iterations for $k = 64$