k-means++: Few More Steps Yield Constant Approximation

Davin Choo [*1] **Christoph Grunau** [*1] **Julian Portmann** [*1] **Václav Rozhoň** [*1]

Abstract

The k-means++ algorithm of Arthur and Vassilvitskii (SODA 2007) is a state-of-the-art algorithm for solving the k-means clustering problem and is known to give an $\mathcal{O}(\log k)$-approximation in expectation. Recently, Lattanzi and Sohler (ICML 2019) proposed augmenting k-means++ with $\mathcal{O}(k \log \log k)$ local search steps to yield a constant approximation (in expectation) to the k-means clustering problem. In this paper, we improve their analysis to show that, for any arbitrarily small constant $\varepsilon > 0$, with only εk additional local search steps, one can achieve a constant approximation guarantee (with high probability in k), resolving an open problem in their paper.

1. Introduction

k-means clustering is an important unsupervised learning task often used to analyze datasets. Given a set P of points in d-dimensional Euclidean space $\mathbb{R}^d$ and an integer k, the task is to partition P into k clusters while minimizing the total cost of the partition. Formally, the goal is to find a set $C \in \mathbb{R}^d$ of k centers minimizing the following objective:

$$\sum_{p \in P} \min_{c \in C} \|p - c\|^2,$$

where points $p \in P$ are assigned to the closest candidate center $c \in C$.

Finding an optimal solution to this objective was proven to be NP-hard (Aloise et al., 2009; Mahajan et al., 2009), and the problem was even shown to be hard to approximate to arbitrary precision (Awasthi et al., 2015; Lee et al., 2017). The currently best known approximation ratio is 6.357 (Ahmadian et al., 2019), while other constant factor approximation algorithms exist (Jain & Vazirani, 2001; Kanungo et al., 2004). For constant dimensions d or constant k, $(1 + \varepsilon)$-approximation algorithms are known (Kumar et al., 2004; Bandyapadhyay & Varadarajan, 2015; Cohen-Addad, 2018; Cohen-Addad et al., 2019; Friggstad et al., 2019). However, these algorithms are mainly of theoretical interest and not known to be efficient in practice.

On the practical side of things, the canonical k-means algorithm (Lloyd, 1982) proved to be a good heuristic. Starting with k initial points (e.g. chosen at random), Lloyd's algorithm iteratively, in an alternating minimization manner, assigns points to the nearest center and updates the centers to be the centroids of each cluster, until convergence. Although the alternating minimization provides no provable approximation guarantee, Lloyd's algorithm never increases the cost of the initial clustering. Thus, one way to obtain theoretical guarantees is to provide Lloyd's algorithm with a provably good initialization.

The k-means++ algorithm (see Algorithm 1) of Arthur and Vassilvitskii (2007) is a well-known algorithm for computing an initial set of k centers with provable approximation guarantees. The initialization is performed by incrementally choosing k initial seeds for Lloyd using D^2-sampling, i.e., sample a point with probability proportional to its squared distance to the closest existing center. They showed that the resultant clustering is an $\mathcal{O}(\log k)$-approximation in expectation. This theoretical guarantee is substantiated by empirical results showing that k-means++ can heavily outperform random initialization, with only a small amount of additional computation time on top of running Lloyd's algorithm. However, lower bound analyses (Brunsch & Röglin, 2013; Bhattacharya et al., 2016) show that there exist inputs where k-means++ is $\Omega(\log k)$-competitive with high probability in k.

Recently, Lattanzi and Sohler (2019) proposed a variant of local search after picking k initial centers via k-means++ (see Algorithm 2): In each step, a new point is sampled with probability proportional to its current cost and used to replace an existing center such as to maximize the cost reduction. If all possible swaps increase the objective cost, the new sampled point is discarded. Following their notation, we refer to this local search procedure as LocalSearch++. They showed that performing

[*] Equal contribution [1] ETH Zürich. Correspondence to: Davin Choo <chood@ethz.ch>, Christoph Grunau <cgrunau@ethz.ch>, Julian Portmann <pjulian@ethz.ch>, Václav Rozhoň <rozhonv@ethz.ch>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

Algorithm 1 k-means++ seeding

```
Input: $P$, $k$, $\ell$
Uniformly sample $p \in P$ and set $C = \{p\}$.
for $i \leftarrow 2, 3, \ldots, k$ do
    Sample $p \in P$ w.p. $\frac{cost(p,C)}{\sum_{q \in P} cost(q,C)}$ and add it to $C$.
end for
```

Algorithm 2 One step of LocalSearch++

```
Input: $P$, $C$
Sample $p \in P$ with probability $\frac{cost(p,C)}{\sum_{q \in P} cost(q,C)}$
$p' = \arg\min_{q \in C} cost(P, C \setminus \{q\} \cup \{p\})$
if $cost(P, C \setminus \{p'\} \cup \{p\}) < cost(P, C)$ then
    $C = C \setminus \{p'\} \cup \{p\}$
end if
return $C$
```

$\mathcal{O}(k \log\log k)$ steps of LocalSearch++ after k-means++ improves the expected approximation factor from $\mathcal{O}(\log k)$ to $\mathcal{O}(1)$, and stated that it is an interesting open question to prove that $\mathcal{O}(k)$ local search steps suffice to obtain a constant factor approximation.

1.1. Our Contribution

In this paper, we answer the open question by Lattanzi and Sohler (2019) in the affirmative. We refine their analysis to show that with only εk additional local search steps, one can achieve an approximation guarantee of $\mathcal{O}(1/\varepsilon^3)$ with probability $1 - \exp\left(-\Omega\left(k^{0.1}\right)\right)$. Compared to (Lattanzi & Sohler, 2019), we improve the number of search steps needed to achieve a constant approximation from $\mathcal{O}(k \log\log k)$ to just εk.

To achieve this result, we go beyond the worst-case analysis of k-means++ that shows that it is $\Omega(\log k)$ approximate in the worst case (Arthur & Vassilvitskii, 2007) (Theorem 3.1); we prove in Lemma 12 that it is always the case that most optimal clusters are approximated up to a constant factor (this goes in the similar spirit as bi-criteria result of Wei (2016)). This enables us to show that a few steps of local search is enough to "fix" these "few bad clusters".

Theorem 1 (Main theorem)**.** *Let $k \in \Omega(1/\varepsilon^{20})$ and $0 < \varepsilon \leq 1$. Suppose we run Algorithm 1 followed by $\ell = \varepsilon k$ steps of Algorithm 2. We have $cost(P, C) \leq \left(10^{30}/\varepsilon^3\right) \cdot cost(P, C^*)$ with probability at least $1 - \exp(-\Omega(k^{0.1}))$.*

1.2. Related Work

Another variant of local search was analyzed by Kanungo et al. (2004): In each step, try to improve by swapping an existing center with an input point. Although they showed that this eventually yields a constant approximation, the number of required steps can be very large.

Under the bicriteria optimization setting, Aggarwal et al. (2009) and Wei (2016) proved that if one over-samples and runs D^2-sampling for $\mathcal{O}(k)$ steps (instead of just k), one can get a constant approximation of the k-means objective with these $\mathcal{O}(k)$ centers. We note that a single step of LocalSearch++ has almost the same asymptotic running time as over-sampling once using D^2-sampling (see Section 3.3 for details) while enforcing the constraint of *exactly* k centers. However, their results are stronger in terms of approximation guarantees: Aggarwal et al. (2009) proved that in $\mathcal{O}(k)$ steps one achieves a $4 + \varepsilon$ approximation to the optimal cost with constant probability, while Wei (2016) proved that after εk more sampling steps one achieves an $\mathcal{O}(1/\varepsilon)$-approximation in expectation.

Other related work include speeding up k-means++ via approximate sampling (Bachem et al., 2016), approximating k-means++ in the streaming model (Ackermann et al., 2012), and running k-means++ in a distributed setting (Bahmani et al., 2012).

1.3. Our Method, in a Nutshell

Lattanzi and Sohler showed that given any clustering with approximation ratio of at least 500, a single step of LocalSearch++ improves the cost by a factor of $1 - 1/(100k)$, with constant probability. In general, one cannot hope to asymptotically improve their bound[1]. Instead, our improvement comes from structural insights on solutions provided by Algorithm 1 and Algorithm 2. We argue that if we have a bad approximation at any point in time, the next step of Algorithm 2 drastically improves the cost with a positive constant probability.

To be more specific, consider an optimal clustering OPT. We prove that throughout the course of Algorithm 2, most centroids of OPT clusters have a candidate center in the current solution close to it. Through a chain of technical lemmas, à la (Lattanzi & Sohler, 2019), we conclude that the new sampled point is close to the centroid of one of the (very few) costly OPT clusters with constant probability. Then, we argue that there is an existing candidate that can be swapped with the newly sampled point to improve the solution quality substantially. Putting everything together, we get that the solution improves by a factor of $1 - \Theta(\sqrt[3]{\alpha}/k)$ with constant probability in a single LocalSearch++ step, where α is the approximation factor of the current solution. This improved multiplicative cost reduction suffices to prove our main result.

In Section 2, we introduce notation and crucial definitions,

[1] Consider a $(k-1)$-dimensional simplex with n/k points at each corner and a clustering with all k centers in the same corner. Swapping one center to another corner of the simplex improves the solution by a factor of $1 - \Theta(1/k)$. However, we do not expect to get such a solution after running the k-means++ algorithm.

along with several helpful lemmas. In Section 3, we walk the reader through our proof while deferring some lengthy proofs to the supplementary material.

2. Preliminaries

Let P be a set of points in $\mathbb{R}^d$. For two points $p, q \in \mathbb{R}^d$, let $\|p - q\|$ be their Euclidean distance. We denote $C \subseteq P$ as the set of candidate centers and $C^* = OPT$ as the centers of a (fixed) optimal solution, where $|C^*| = k$. Note that a center $c^* \in C^*$ may *not* be a point from P while all candidates $c \in C$ are actual points from P. For $c^* \in C^*$, the set Q_{c^*} denotes the points in P that OPT assigns to c^*. We define $cost(P, C) = \sum_{p \in P} \min_{c \in C} \|p - c\|^2$ as the cost of centers C, where $cost(P, C^*)$ is the cost of an optimal solution. When clear from context, we also refer to the optimal cost as OPT. For an arbitrary set of points Q, we denote their centroid by $\mu_Q = (1/|Q|) \cdot \sum_{q \in Q} q$. Note that μ_Q may not be a point from Q.

For the sake of readability, we drop the subscript Q when there is only one set of points in discussion and we drop braces when describing singleton sets in $cost(\cdot, \cdot)$. We will also ignore rounding issues as they do not play a critical role asymptotically.

We now define D^2-sampling introduced in k-means++.

Definition 2 (D^2-sampling)**.** *Given a set $C \subseteq P$ of candidate centers, we sample a point $p \in P$ with probability $\boldsymbol{P}[p] = cost(p, C) / \sum_{p \in P} cost(p, C)$.*

The following folklore lemma describes an important property of the cost function. This is analogous to the bias-variance decomposition in machine learning and to the parallel axis theorem in physics (Aggarwal et al., 2009). As the variable naming suggests, we will use it with Q being an OPT center and c being a candidate center.

Lemma 3. *Let $Q \subseteq P$ be a set of points. For any point $c \in P$ (possibly not in Q),*

$$cost(Q, c) = |Q| \cdot \|c - \mu_Q\|^2 + cost(Q, \mu_Q)$$

To have a finer understanding of the cluster structure, we define the notions of *settled* and *approximate* clusters. Consider an arbitrary set of points $Q \subseteq P$ (e.g. some cluster of OPT). We define

$$R_{Q,\beta} = \{q \in Q : \|q - \mu_Q\|^2 \leq (\beta/|Q|) \cdot cost(Q, \mu_Q)\}$$

as the subset of points in Q that are within a certain radius from μ_Q (i.e. "close" with respect to β). As β decreases, the condition becomes stricter and the set $R_{Q,\beta}$ shrinks.

Definition 4 (β-settled)**.** *An OPT cluster Q is β-settled if $R_\beta \cap C \neq \emptyset$. That is, there is a candidate center $c \in Q$ with distance at most $(\beta/|Q|) \cdot cost(Q, \mu_Q)$ from μ_Q.*

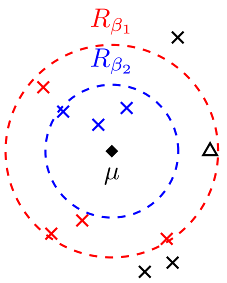

Figure 1: × are points in set Q with centroid μ (which may *not* be a point from Q). For $\beta_1 > \beta_2$, $R_{\beta_2} \subseteq R_{\beta_1} \subseteq Q$. △ represents a candidate center $c \in C$ that does *not* belong to Q. Since $c \notin Q$, cluster Q is *not* β_1-settled even though $\|c - \mu_Q\|^2 \leq (\beta_1/|Q|) \cdot cost(Q, \mu_Q)$.

Definition 5 (α-approximate)**.** *An OPT cluster Q is α-approximate if $cost(Q, C) \leq \alpha \cdot cost(Q, \mu_Q)$.*

Intuitively, a β-settled cluster Q has small $cost(Q, C)$. As settled-ness requires a candidate $c \in C$ to belong to cluster Q, an unsettled cluster Q could have small $cost(Q, C)$. See Fig. 1 for an illustration of Q, μ_Q, $R_{Q,\beta}$ and β-settled. We now relate the definitions of settled and approximate.

Lemma 6. *Suppose Q is a cluster of OPT that is β-settled. Then, $cost(Q, C) \leq (\beta + 1) \cdot cost(Q, \mu)$. In other words, β-settled implies $(\beta + 1)$-approximate.*

Proof. For any β-settled cluster Q, there is some candidate center $c \in C$ in R_β, so

$$\begin{aligned} cost(Q, C) &\leq cost(Q, c) = |Q| \cdot \|c - \mu\|^2 + cost(Q, \mu) \\ &\leq (\beta + 1) \cdot cost(Q, \mu) = (\beta + 1) \cdot cost(Q, C^*) \end{aligned}$$

□

It is also useful to consider the contrapositive of Lemma 6.

Corollary 7. *Let Q be a cluster of OPT. If $cost(Q, C) > (\beta + 1) \cdot cost(Q, \mu)$, then $\|c - \mu\|^2 > (\beta/|Q|) \cdot cost(Q, \mu)$ for any candidate center $c \in C$. That is, Q is β-unsettled.*

In our analysis, we will prove statements about clusters being settled for general values of β. However, there are uncountably many possible β's and therefore we cannot do a union bound over all possible choices of β. Using a similar idea to ε-net arguments, we will discretize the set of β's into a sufficiently small finite set of *legal* values.

Definition 8 (Legal β values)**.** *A parameter β is* legal *if $\beta \in \mathcal{B} = \{2^i : i \in \{3, 4, \ldots, 0.3 \cdot \log k\}\}$. In particular, this implies that all legal β are at least 8 and at most $k^{0.3}$.*

3. Analysis

In this section, we present the key ideas of our proof while deferring some details to the supplementary material.

Following the proof outline of Lattanzi and Sohler (2019), we perform a more intricate analysis of `LocalSearch++`. Their key lemma shows that, with constant probability, the cost of the solution decreases by a factor of $1 - 1/(100k)$ after one local search step.

Lemma 3 in (Lattanzi & Sohler, 2019) Let P be a set of points and C be a set of centers with $cost(P, C) > 500\, OPT$. Denote the updated centers by $C' = \texttt{LocalSearch++}(P, C)$. Then, with probability $1/1000$, $cost(P, C') \leq (1 - 1/(100k))\, cost(P, C)$.

The above lemma implies that we expect the cost of the current solution to drop by a constant factor after $\mathcal{O}(k)$ `LocalSearch++` steps (unless we already have a constant approximation of the optimum). Since we start with a solution that is an $\mathcal{O}(\log k)$-approximation in expectation, we expect that after $\mathcal{O}(k \log\log k)$ iterations, the cost of our solution drops to a constant. This yields the main theorem of (Lattanzi & Sohler, 2019).

Theorem 1 in (Lattanzi & Sohler, 2019) Let P be a set of points and C be the output of k-`means++` followed by at least $100000\, k \log\log k$ many local search steps. Then, we have $\mathbf{E}[cost(P, C)] \in \mathcal{O}(cost(P, C^*))$. The running time of the algorithm is $\mathcal{O}(dnk^2 \log\log k)$.

Our improvements rely on the following structural observation: After running k-`means++`, most of the clusters of the optimal solution are already "well approximated" with high probability in k.

3.1. Structural Analysis

In this subsection, we study the event of sampling a point from a β-unsettled cluster and making it β-settled. This allows us to prove concentration results about the number of β-settled clusters, which we will use in the next subsection.

Suppose α is the current approximation factor. The result below states that with good probability, the new sampled point is from a cluster that is currently badly approximated.

Lemma 9. *Suppose that $cost(P, C) = \alpha \cdot cost(P, C^*)$ and we D^2-sample a point $p \in P$. Consider some fixed $\beta \geq 1$. Then, with probability at least $1 - \beta/\alpha$, the sampled point p is from a cluster Q with $cost(Q, C) \geq \beta \cdot cost(Q, \mu_Q)$.*

Proof. Let $\widetilde{Q}$ be the union of all clusters Q such that $cost(Q, C) < \beta \cdot cost(Q, \mu_Q)$. By definition, D^2-sampling will pick a point from $\widetilde{Q}$ with probability at most $cost(\widetilde{Q}, C)/cost(P, C) \leq \beta/\alpha$. □

Similar to most work[2] on k-`means++`, we need a sampling lemma stating that if we sample a point within a cluster Q (according to D^2 weights of points in Q), then the sampled point will be relatively close to μ_Q with good probability.

Lemma 10. *Suppose that Q is an OPT cluster with $cost(Q, C) \geq \beta \cdot cost(Q, \mu_Q)$ for some $\beta \geq 4$ and we D^2-sample a point $p \in Q$. Then, with probability at least $1 - 6/\sqrt{\beta}$, Q becomes $(\beta - 1)$-settled. That is, $\|p - \mu_Q\|^2 \leq ((\beta - 1)/|Q|) \cdot cost(Q, \mu_Q)$ and $cost(Q, p) \leq \beta \cdot cost(Q, \mu_Q)$.*

Proof. Let $\beta' \geq \beta$ be the *exact* approximation factor. i.e. $cost(Q, C) = \beta' \cdot cost(Q, \mu)$. We define sets Q_{in} and P'_{in}:

$$Q_{\text{in}} = \left\{ q \in Q : \|q - \mu_Q\| \leq \sqrt{\frac{\beta - 2}{|Q|} cost(Q, \mu_Q)} \right\}$$

$$P'_{\text{in}} = \left\{ p \in P : \|p - \mu_Q\| \leq \sqrt{\frac{\beta' - 2}{|Q|} cost(Q, \mu_Q)} \right\}$$

By definition, we have $Q_{\text{in}} \subseteq Q$ and $Q_{\text{in}} \subseteq P'_{\text{in}}$. However, $P'_{\text{in}} \subseteq Q$ does not hold in general.

Lemma 3 tells us that $P'_{\text{in}} \cap C = \emptyset$. Otherwise $\beta' \cdot cost(Q, \mu_Q) = cost(Q, C) \leq (\beta' - 1) \cdot cost(Q, \mu_Q)$, which is a contradiction. Furthermore, it holds that $|Q \setminus Q_{\text{in}}| \leq |Q|/(\beta - 2)$. Otherwise $cost(Q, \mu_Q) > (|Q|/(\beta - 2)) \cdot ((\beta - 2)/|Q|) \cdot cost(Q, \mu_Q)$, which is a contradiction. Hence, $|Q_{\text{in}}| \geq (1 - 1/(\beta - 2)) \cdot |Q|$.

Let $d_i = \|q_i - \mu_Q\|$ be the distance of the i-th point of Q_{in} from μ_Q, so $\sum_{i=1}^{|Q_{\text{in}}|} d_i^2 \leq cost(Q, \mu_Q)$. By the Cauchy-Schwarz inequality, we have $\sum_{i=1}^{|Q_{\text{in}}|} d_i \leq \sqrt{|Q_{\text{in}}| \cdot \sum_{i=1}^{|Q_{\text{in}}|} d_i^2} \leq \sqrt{|Q| \cdot cost(Q, \mu_Q)}$. Since $P'_{\text{in}} \cap C = \emptyset$, triangle inequality tells us that $\sqrt{cost(q_i, C)} \geq \sqrt{cost(\mu_Q, C)} - \sqrt{cost(q_i, \mu_Q)} \geq \sqrt{((\beta' - 2)/|Q|) \cdot cost(Q, \mu_Q)} - d_i$ for each point $q_i \in Q_{\text{in}}$. Thus,

$$\begin{aligned} cost(Q_{\text{in}}, C) &= \sum_{i=1}^{|Q_{\text{in}}|} cost(q_i, C) \\ &\geq \sum_{i=1}^{|Q_{\text{in}}|} \left(\sqrt{\frac{\beta' - 2}{|Q|} cost(Q, \mu_Q)} - d_i \right)^2 \\ &\geq \sum_{i=1}^{|Q_{\text{in}}|} \frac{\beta' - 2}{|Q|} cost(Q, \mu_Q) - 2d_i \sqrt{\frac{\beta' - 2}{|Q|} cost(Q, \mu_Q)} \\ &= \frac{|Q_{\text{in}}|}{|Q|} (\beta' - 2) cost(Q, \mu_Q) \\ &\quad - 2\sqrt{\frac{\beta' - 2}{|Q|} cost(Q, \mu_Q)} \cdot \sum_{i=1}^{|Q_{\text{in}}|} d_i \end{aligned}$$

[2] Cf. Lemma 2 in (Arthur & Vassilvitskii, 2007), Lemma 5 of (Aggarwal et al., 2009), Lemma 6 of (Lattanzi & Sohler, 2019).

$$\begin{aligned}
&\geq \left(1-\frac{1}{\beta-2}\right)(\beta'-2)\,cost(Q,\mu_Q)\\
&\quad -2\sqrt{\frac{\beta'}{|Q|}cost(Q,\mu_Q)}\cdot\sqrt{|Q|cost(Q,\mu_Q)}\\
&=\left(\left(1-\frac{1}{\beta-2}\right)(\beta'-2)-2\sqrt{\beta'}\right)cost(Q,\mu_Q)\\
&=\left(\left(1-\frac{1}{\beta-2}\right)\left(1-\frac{2}{\beta'}\right)-\frac{2}{\sqrt{\beta'}}\right)cost(Q,C)\\
&\geq\left(1-\frac{2+2+2}{\sqrt{\beta}}\right)cost(Q,C)\\
&=\left(1-\frac{6}{\sqrt{\beta}}\right)cost(Q,C)
\end{aligned}$$

Hence, the probability that the sampled point p is taken from Q_{in} is at least $1-6/\sqrt{\beta}$. Having sampled a point $p \in Q$ with $\|p-\mu_Q\|^2 \leq ((\beta-1)/|Q|)\cdot cost(Q,\mu_Q)$, Lemma 3 tells us that the cost of cluster Q is at most $\beta \cdot cost(Q,\mu_Q)$. □

Corollary 11. *Fix $\alpha \geq 10$ such that $cost(P,C) = \alpha \cdot OPT$ and let $1 < \beta \leq \alpha^{2/3}$. Suppose that we D^2-sample a new point $p \in P$. Then, with probability at least $1-8/\sqrt{\beta}$, the sampled point p is from a β-unsettled cluster and this cluster becomes β-settled.*

Proof. Lemma 9 tells us that, with probability at least $1-(2\beta)/\alpha$, we sample from an OPT cluster Q with $cost(Q,C) \geq (2\beta)\cdot cost(Q,\mu_Q)$. As $cost(Q,C) > (\beta+1)\cdot cost(Q,\mu_Q)$, Corollary 7 implies that Q is β-unsettled. According to Lemma 10, Q becomes β-settled with probability at least $1-6/\sqrt{\beta+1} \geq 1-6/\sqrt{\beta}$. As $\beta \leq \alpha^{2/3}$, the probability of the first event is at least $1-(2\beta)/\alpha \geq 1-(2\beta)/\beta^{3/2} = 1-2/\sqrt{\beta}$. Thus, the joint event of sampling from a β-unsettled cluster and making it β-settled happens with probability at least $1-8/\sqrt{\beta}$. □

We can now use Corollary 11, together with a Chernoff Bound, to upper-bound the number of β-unsettled clusters. First, we show that with high probability, k-means++ leaves only a small number of clusters β-unsettled for every legal $\beta \leq \alpha^{2/3}$. Then, we show that this property is maintained throughout the course of the local search.

Lemma 12. *After running k-means++ (for k steps) and $\ell \leq k$ steps of LocalSearch++, let C denote the set of candidate centers and $\alpha \geq 1$ be the approximation factor. Then, with probability at least $1-\exp(-\Omega(k^{0.1}))$, there are at most $(30k)/\sqrt{\beta}$ clusters that are β-unsettled, for any legal $\beta \leq \alpha^{2/3}$.*

Recall that every legal β is smaller than $k^{0.3}$, as for larger β one cannot obtain strong concentration results. For reasonably small α, Lemma 12 allows us to conclude that there are at most $\mathcal{O}(k/\sqrt{\alpha^{2/3}}) = O(k/\sqrt[3]{\alpha})$ clusters that are $\alpha^{2/3}$-unsettled, with high probability in k. Conditioned on this event, we can expect a stronger multiplicative improvement in one iteration of LocalSearch++ compared to Lemma 3 of (Lattanzi & Sohler, 2019).

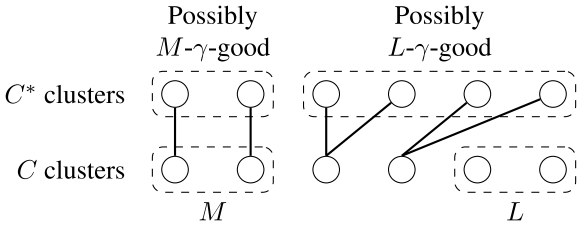

Figure 2: Let $k = 6$. The top row represents the k OPT centers C^*. The bottom row represents the k candidate centers C. Each OPT center is connected to the closest candidate center by a line. Observe that M and L are subsets of C, where some candidate centers might be in neither M nor L, and that γ-goodness is defined on the OPT centers. This example shows a tight case for Observation 14.

3.2. One Step of LocalSearch++

Given the structural analysis of the previous section, we can now analyze the LocalSearch++ procedure. First, we will identify clusters whose removal will not significantly increase the current cost, thus making them good candidates for swapping with the newly sampled center.

To that end, we define subsets of *matched* and *lonely* candidate centers $M \subseteq C$ and $L \subseteq C$. The notion of lonely centers came from Kanungo et al. (2004). To describe the *same* subsets, Lattanzi and Sohler (2019) used the notation H and L, while we use M and L. For an illustration of these definitions, see Fig. 2.

Definition 13 (M and L candidates)**.** *We assign OPT centers $c^* \in C^*$ to candidate centers $c \in C$, and define the notion of matched (M) and lonely (L) on candidates based on assignment outcome. For each $c^* \in C^*$, assign c^* to the closest $c \in C$, breaking ties arbitrarily. We say candidate $c \in C$ is* matched *if there is* exactly one *$c^* \in C^*$ assigned to it and we call c^* the mate of c. We say candidate $c \in C$ is* lonely *if there is* no *$c^* \in C^*$ assigned to it.*

We define $M \subseteq C$ as the set of matched candidates and $L \subseteq C$ as the set of lonely candidates. We sometimes overload notation and write $(c, c^*) \in M$ if $c \in C$ is a matched candidate center with mate $c^* \in C^*$.

Observation 14. *Since $|C| = |C^*| = k$, a counting argument tells us that $k - |M| \leq 2|L|$.*

We now define reassignment costs for candidate centers

$c \in M \cup L \subseteq C$ and the notion of γ-good OPT centers[3]. Informally, OPT center $c^* \in C^*$ is γ-good if selecting a random point in Q_{c^*} and removing a suitable candidate $c \in C$ reduces a "sufficient" fraction of the current clustering cost with a constant probability.

Definition 15 (Reassignment costs)**.**
If $(c, c^*) \in M$*,*

$$\mathtt{reassign}(P, C, c) = cost(P \setminus Q_{c^*}, C \setminus \{c\}) - cost(P \setminus Q_{c^*}, C)$$

If $c \in L$*,*

$$\mathtt{reassign}(P, C, c) = cost(P, C \setminus \{c\}) - cost(P, C)$$

We will use the following lemma about reassignment costs, proven in Lemma 4 of (Lattanzi & Sohler, 2019).

Lemma 16. *For* $c \in M \cup L$*, with* P_c *as the points assigned to* c*,*

$$\mathtt{reassign}(P, C, c) \leq \frac{21}{100} cost(P_c, C) + 24 cost(P_c, C^*)$$

Definition 17 (M-γ-good and L-γ-good)**.**
We say that $c^* \in C^* \cap M$ *with mate* $c \in C$ *is* M*-*γ*-good if*

$$cost(Q_{c^*}, C) - \mathtt{reassign}(P, C, c) - 100 \cdot cost(Q_{c^*}, c^*) > \frac{\gamma}{10^4 k} \cdot cost(P, C).$$

We say that $c^* \in C^* \setminus M$ *is* L*-*γ*-good if*

$$cost(Q_{c^*}, C) - \min_{c \in L} \mathtt{reassign}(P, C, c) - 100 \cdot cost(Q_{c^*}, c^*) > \frac{\gamma}{10^4 k} \cdot cost(P, C).$$

Claim 18. *Let* Q *be a* M*-*γ*-good or* L*-*γ*-good cluster and we* D^2*-sample a point* $q \in Q$*. Then, with probability at least* $2/5$*, we have* $cost(Q, q) \leq 100 \cdot cost(Q, \mu_Q)$*.*

Proof. Let C denotes the current set of candidate centers. Suppose cluster Q is M-γ-good. Then,

$$cost(Q, C) > \mathtt{reassign}(P, C, c) + 100 \cdot cost(Q, \mu_Q) + \frac{\gamma}{10^4 k} \cdot cost(P, C) \geq 100 \cdot cost(Q, \mu_Q)$$

By Lemma 10, we have $cost(Q, q) \leq 100 \cdot cost(Q, \mu_Q)$ with probability at least $1 - 6/\sqrt{100} = 2/5$. The same argument holds when Q is an L-γ-good cluster by applying the definition of L-γ-good instead. □

Conditioned on our main structural insight (Lemma 12), we sample a point from an M-$\sqrt{\beta}$-good or L-$\sqrt{\beta}$-good cluster Q with constant probability, for every legal $\beta \in \mathcal{B}$ such that $4 \leq \beta \leq \alpha^{2/3}$ and $\alpha \geq 10^9$. When this happens, the sampled point s satisfies $cost(Q, s) \leq 100 \cdot cost(Q, \mu_Q)$ with constant probability. In that case, the definition of M-$\sqrt{\beta}$-good and L-$\sqrt{\beta}$-good implies the existence of a candidate $t \in C$ with $cost(P, C \setminus \{t\} \cup \{s\}) \leq (1 - \frac{\sqrt{\beta}}{10^4 k}) \cdot cost(P, C)$, so `LocalSearch++` makes progress.

Similar to the analysis of Lattanzi and Sohler (2019), we partition the space of possible events into whether $\sum_{(c,c^*) \in M} cost(Q_{c^*}, C) \geq cost(P, C)/2$, or not. In each case, we argue that the probability of sampling a point contained in a M-$\sqrt{\beta}$-good or L-$\sqrt{\beta}$-good cluster happens with a positive constant probability for a suitable legal β. We first refine Lemma 5 of (Lattanzi & Sohler, 2019).

Lemma 19. *Suppose* $2 \cdot \sum_{(c,c^*) \in M} cost(Q_{c^*}, C) \geq cost(P, C) = \alpha \cdot cost(P, C^*)$ *for* $\alpha \geq 10^9$*. Let* $4 \leq \beta \leq \alpha^{2/3}$ *be arbitrary. If there are at most* $(30k)/\sqrt{\beta}$ *clusters that are* β*-unsettled, then*

$$\sum_{\substack{(c,c^*) \in M, \\ c^* \in M\text{-}\sqrt{\beta}\text{-good}}} cost(Q_{c^*}, C) \geq \frac{1}{500} \cdot cost(P, C).$$

Lemma 19 tells us that if points in M have sufficiently large probability mass, then points in M-$\sqrt{\beta}$-good clusters hold a constant fraction of the total probability mass.

Proof. We show that the probability mass is large by upper bounding the probability mass on its negation. To do this, we partition the summation of $c^* \notin M$-$\sqrt{\beta}$-good into β-settled and β-unsettled. We denote

$$\mathcal{A} = \left\{ (c, c^*) \in M, c^* \notin M\text{-}\sqrt{\beta}\text{-good}, c^* \text{ is } \beta\text{-settled} \right\}$$

$$\mathcal{B} = \left\{ (c, c^*) \in M, c^* \notin M\text{-}\sqrt{\beta}\text{-good}, c^* \text{ is } \beta\text{-unsettled} \right\}$$

From Lemma 6, we know that C pays no more than $(\beta + 1) \cdot cost(P, C^*)$ for all β-settled clusters. So,

$$\sum_{\mathcal{A}} cost(Q_{c^*}, C) \leq (\beta + 1) \cdot cost(P, C^*) \leq (\alpha^{2/3} + 1) \cdot cost(P, C^*) \leq \frac{2\alpha^{2/3}}{\alpha} \cdot cost(P, C) \leq \frac{1}{500} \cdot cost(P, C)$$

To bound $\sum_{\mathcal{B}} cost(Q_{c^*}, C)$, recall that P is the set of *all* points and $Q_{c^*} \subseteq P$ for any $c^* \in C^*$.

$$\sum_{\mathcal{B}} cost(Q_{c^*}, C)$$

[3] For clarity, we define using γ. Later, we set $\gamma = \sqrt{\beta}$.

$$
\begin{aligned}
&\leq \sum_{\mathcal{B}} \Big(\texttt{reassign}(P, C, c) + 100 \cdot cost(Q_{c^*}, c^*) \\
&\quad + \frac{\sqrt{\beta}}{10^4 k} \cdot cost(P, C) \Big) && (\star) \\
&\leq \left(\sum_{\mathcal{B}} \texttt{reassign}(P, C, c) \right) + 100 \cdot cost(P, C^*) \\
&\quad + \frac{30}{10^4} \cdot cost(P, C) && (\dagger) \\
&\leq \frac{21}{100} \cdot cost(P, C) + 24 \cdot cost(P, C^*) \\
&\quad + 100 \cdot cost(P, C^*) + \frac{30}{10^4} \cdot cost(P, C) && (\ddagger) \\
&\leq \frac{250}{1000} \cdot cost(P, C) && (*)
\end{aligned}
$$

(Legend) $(\star)$: Definition 17; $(\dagger)$: because there are at most $\frac{30k}{\sqrt{\beta}}$ clusters that are β-unsettled; $(\ddagger)$: Lemma 16; $(*)$: $cost(P, C) \geq 10^9 \cdot cost(P, C^*)$

Thus,

$$
\begin{aligned}
&\sum_{\substack{(c,c^*) \in M, \\ c^* \,\in\, M\text{-}\sqrt{\beta}\text{-good}}} cost(Q_{c^*}, C) \\
&\geq \left(\frac{1}{2} - \frac{1}{500} - \frac{250}{1000} \right) \cdot cost(P, C) \geq \frac{1}{500} \cdot cost(P, C)
\end{aligned}
$$

□

Using the same structural insight on β-unsettled clusters, we now refine Lemma 7 of (Lattanzi & Sohler, 2019). Abusing notation, we use $C^* \setminus M$ to denote the set of optimal cluster centers which don't have a mate. That is, the point $c \in C$ which c^* is assigned to is assigned to has more than one center of C^* assigned to it.

Lemma 20. *Suppose* $2 \cdot \sum_{(c,c^*) \in M} cost(Q_{c^*}, C) < cost(P, C) = \alpha \cdot cost(P, C^*)$ *for* $\alpha \geq 10^9$. *Let* $4 \leq \beta \leq \alpha^{2/3}$ *be arbitrary. If there are at most* $(30k)/\sqrt{\beta}$ *clusters that are* β*-unsettled, then*

$$
\sum_{\substack{c^* \in C^* \setminus M, \\ c^* \,\in\, L\text{-}\sqrt{\beta}\text{-good}}} cost(Q_{c^*}, C) \geq \frac{1}{500} \cdot cost(P, C).
$$

Lemma 20 tells us that if points in $C^* \setminus M$ have sufficiently large probability mass, then points in L-$\sqrt{\beta}$-good clusters hold a constant fraction of the total probability mass.

With Lemma 19 and Lemma 20, we can now refine Lemma 3 from (Lattanzi & Sohler, 2019).

Lemma 21. *Suppose we have a clustering* C *with* $cost(P, C) = \alpha \cdot cost(P, C^*)$ *for some* $\alpha \geq 10^9$. *Assume that for each legal* β*, where* $\beta \leq \alpha^{2/3}$*, there are at most* $(30k)/\sqrt{\beta}$ *clusters that are* β*-unsettled. If we update* C *to* C' *in one* `LocalSearch++` *iteration, we have with probability at least* $1/2000$*:*

$$
cost(P, C') \leq \left(1 - \frac{\min\{\sqrt[3]{\alpha},\, k^{0.15}\}}{2 \cdot 10^4 k} \right) \cdot cost(P, C)
$$

Proof. Pick a legal $\beta \in \mathcal{B}$ such that $\frac{1}{2} \min\left\{k^{0.3}, \alpha^{2/3}\right\} \leq \beta < \min\left\{k^{0.3}, \alpha^{2/3}\right\}$. We define M and L candidate centers as in Definition 13 and consider the following two cases separately:

1. $\sum_{(c,c^*) \in M} cost(Q_{c^*}, C) \geq \frac{1}{2} \cdot cost(P, C)$
2. $\sum_{(c,c^*) \in M} cost(Q_{c^*}, C) < \frac{1}{2} \cdot cost(P, C)$

Let q be the D^2-sampled point and $c \in C$ be some current candidate center, which we will define later in each case. In both cases (1) and (2), we will show that the pair of points (q, c) will fulfill the condition $cost(P, C \cup \{q\} \setminus \{c\}) \leq \left(1 - \sqrt{\beta} / \left(10^4 \cdot k\right)\right) \cdot cost(P, C)$ with some constant probability. The claim follows since the algorithm takes the $c \in C$ that decreases the cost the most and swaps it with the D^2-sampled point q.

Case (1): $\sum_{(c,c^*) \in M} cost(Q_{c^*}, C) \geq \frac{1}{2} \cdot cost(P, C)$

Lemma 19 tells us that we sample from a M-$\sqrt{\beta}$-good cluster with probability at least $1/500$. Denote this cluster as Q_{c^*}. Then, by Claim 18, the D^2-sampled point q satisfies $cost(Q_{c^*}, q) \leq 100 \cdot cost(Q_{c^*}, \mu_{Q_{c^*}})$ with probability at least $2/5$. Jointly, with probability at least $2/2500$, we D^2-sampled a "good" point $q \in Q_{c^*}$ where $(c, c^*) \in M$ and $c^* \in M$-$\sqrt{\beta}$-good, so

$$
\begin{aligned}
&cost(P, C \cup \{q\} \setminus \{c\}) \\
&= cost(P, C) - (cost(P, C) - cost(P, C \cup \{q\} \setminus \{c\})) \\
&\leq cost(P, C) - \Big((cost(P \setminus Q_{c^*}, C) + cost(Q_{c^*}, C)) \\
&\quad - (cost(P \setminus Q_{c^*}, C \setminus \{c\}) + cost(Q_{c^*}, q)) \Big) \\
&= cost(P, C) - \Big(cost(Q_{c^*}, C) - (cost(P \setminus Q_{c^*}, C \setminus \{c\}) \\
&\quad - cost(P \setminus Q_{c^*}, C)) - cost(Q_{c^*}, q) \Big) \\
&\leq cost(P, C) - \Big(cost(Q_{c^*}, C) - \texttt{reassign}(P, C, c) \\
&\quad - 100 \cdot cost(Q_{c^*}, \mu_{Q_{c^*}}) \Big) \\
&\leq cost(P, C) - \frac{\sqrt{\beta}}{10^4 \cdot k} \cdot cost(P, C) \\
&= \left(1 - \frac{\sqrt{\beta}}{10^4 \cdot k} \right) \cdot cost(P, C) \\
&\leq \left(1 - \frac{\min\{\sqrt[3]{\alpha}, k^{0.15}\}}{2 \cdot 10^4 \cdot k} \right) \cdot cost(P, C)
\end{aligned}
$$

Case (2): $\sum_{(c,c^*)\in M} cost(Q_{c^*}, C) < \frac{1}{2} \cdot cost(P, C)$

This is the same as Case (1), but we use Lemma 20 instead of Lemma 19. □

From this lemma, we can conclude that if the current approximation factor is very high, we drastically decrease it within just a few steps. In particular, we can show that if we start with an approximation guarantee that is no worse than $\exp(k^{0.1})$, we can decrease it to just $O(1)$ within εk steps, with probability $1 - \exp(-\Omega(k^{0.1}))$. By Markov's inequality, we know that the probability of having an approximation guarantee that is worse than $\exp(k^{0.1})$ is at most $\exp(-\Omega(k^{0.1}))$. Our main theorem[4] now follows:

Theorem 1 (Main theorem)**.** *Let $k \in \Omega(1/\varepsilon^{20})$ and $0 < \varepsilon \leq 1$. Suppose we run Algorithm 1 followed by $\ell = \varepsilon k$ steps of Algorithm 2. We have $cost(P, C) \leq \left(10^{30}/\varepsilon^3\right) \cdot cost(P, C^*)$ with probability at least $1 - \exp(-\Omega(k^{0.1}))$.*

3.3. Concluding Remarks

Expectation versus high probability An approximation guarantee in expectation only implies (via Markov inequality) that with a constant probability we get a constant approximation. So, our result is stronger as we get a constant approximation of the optimum cost with a probability of at least $1 - \exp(-\Omega(k^{0.1}))$. To recover a guarantee in expectation, we can run the algorithm twice[5]: Let C_1 be the solution obtained by running k-means++ plus LocalSearch++, let C_2 be the output of another independent run of k-means++, and let $\mathcal{E}$ be the event that LocalSearch++ outputs an $\mathcal{O}(1)$-approximation. Then, the expected cost of $\min\{cost(C_1), cost(C_2)\}$ is

$$\begin{aligned}
&\mathbf{E}[\min\{cost(C_1), cost(C_2)\}] \\
\leq\ & \Pr[\mathcal{E}] \cdot \mathcal{O}(1) + (1 - \Pr[\mathcal{E}]) \cdot \mathbf{E}[cost(C_2)] \\
\leq\ & \left(1 - \exp(-\Omega(k^{0.1}))\right) \cdot \mathcal{O}(1) \\
& + \exp(-\Omega(k^{0.1})) \cdot \mathcal{O}(\log k) \\
\in\ & \mathcal{O}(1)
\end{aligned}$$

Running Time On a d-dimensional data set consisting of n data points, a naive implementation of k-means++ has time complexity $\mathcal{O}(dnk^2)$ and space complexity $\mathcal{O}(dn)$. This running time can be improved to $\mathcal{O}(dnk)$ if each data point tracks its distance to the current closest candidate center in C. This is because D^2-sampling and subsequent updating this data structure can be done in $\mathcal{O}(dn)$ time for each iteration of k-means++.

LocalSearch++ can be implemented in a similar manner where each data point remembers its distance to the closest two candidate centers. Lattanzi and Sohler (Lattanzi & Sohler, 2019) argue that if LocalSearch++ deletes clusters with an average size of $\mathcal{O}(n/k)$, then an iteration of LocalSearch++ can be performed in an amortized running time of $\mathcal{O}(dn)$.

However, in the worst case, each iteration of LocalSearch++ can still take $\mathcal{O}(dnk)$ time. A way to provably improve the worst case complexity is to use more memory. With $\mathcal{O}(dnk)$ space, each data point can store distances to all k centers in a binary search tree. Then, each step can be implemented in $\mathcal{O}(dn \log k)$ time, as updating a binary search tree requires $\mathcal{O}(\log k)$ time.

4. Acknowledgements

We are grateful to Zalan Borsos, Mohsen Ghaffari, Saeed Ilchi, and Andreas Krause for their help and discussing this problem with us. In particular, we thank Mohsen Ghaffari for giving us feedback on previous versions of this paper. We also thank the referees for their useful feedback and suggestions.

References

Ackermann, M. R., Märtens, M., Raupach, C., Swierkot, K., Lammersen, C., and Sohler, C. Streamkm++ a clustering algorithm for data streams. *Journal of Experimental Algorithmics (JEA)*, 17:2–1, 2012.

Aggarwal, A., Deshpande, A., and Kannan, R. Adaptive sampling for k-means clustering. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 15–28. Springer, 2009.

Ahmadian, S., Norouzi-Fard, A., Svensson, O., and Ward, J. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, (0):FOCS17–97, 2019.

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Awasthi, P., Charikar, M., Krishnaswamy, R., and Sinop, A. K. The hardness of approximation of euclidean k-means. *arXiv preprint arXiv:1502.03316*, 2015.

[4] The cube-root of α in Lemma 21 is precisely why we obtain an approximation factor of $\mathcal{O}(1/\varepsilon^3)$ after εk LocalSearch++ steps, with high probability in k.

[5] It is not unusual to run k-means++ multiple times in practice. e.g. See documentation of sklearn.cluster.KMeans.

Bachem, O., Lucic, M., Hassani, H., and Krause, A. Fast and provably good seedings for k-means. In *Advances in neural information processing systems*, pp. 55–63, 2016.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. Scalable k-means++. *Proceedings of the VLDB Endowment*, 5(7):622–633, 2012.

Bandyapadhyay, S. and Varadarajan, K. On variants of k-means clustering. *arXiv preprint arXiv:1512.02985*, 2015.

Bhattacharya, A., Jaiswal, R., and Ailon, N. Tight lower bound instances for k-means++ in two dimensions. *Theoretical Computer Science*, 634:55–66, 2016.

Brunsch, T. and Röglin, H. A bad instance for k-means++. *Theoretical Computer Science*, 505:19–26, 2013.

Cohen-Addad, V. A fast approximation scheme for low-dimensional k-means. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 430–440. SIAM, 2018.

Cohen-Addad, V., Klein, P. N., and Mathieu, C. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. *SIAM Journal on Computing*, 48(2):644–667, 2019.

Friggstad, Z., Rezapour, M., and Salavatipour, M. R. Local search yields a ptas for k-means in doubling metrics. *SIAM Journal on Computing*, 48(2):452–480, 2019.

Jain, K. and Vazirani, V. V. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296, 2001.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

Kumar, A., Sabharwal, Y., and Sen, S. A simple linear time (1+/spl epsiv/)-approximation algorithm for k-means clustering in any dimensions. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 454–462. IEEE, 2004.

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

Lee, E., Schmidt, M., and Wright, J. Improved and simplified inapproximability for k-means. *Information Processing Letters*, 120:40–43, 2017.

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

Mahajan, M., Nimbhorkar, P., and Varadarajan, K. The planar k-means problem is np-hard. In *International Workshop on Algorithms and Computation*, pp. 274–285. Springer, 2009.

Wei, D. A constant-factor bi-criteria approximation guarantee for k-means++. In *Advances in Neural Information Processing Systems*, pp. 604–612, 2016.